

**COMP2211**

**University of Southampton**

**Software Engineering Group Project**

**Deliverable 4**

**Code Increment 3**

**Version History**

No.	Date	Comments
1	5 <sup>th</sup> May 2023	First Submission

**Group 1 Members**

Name	Email
Gavin Teo Siu Chyi	gsct1c21@soton.ac.uk
Yohith V. Nyanasegaran	yvn1e20@soton.ac.uk
Lim Shi Xun	sx11c21@soton.ac.uk
Ang Jing Ru	jra1c21@soton.ac.uk

# Table of Contents

<b>1</b>	<b><i>Application</i></b>	<b>1</b>
1.1	<b>Product Value</b>	1
1.2	<b>Application Details</b>	2
<b>2</b>	<b><i>Extensions</i></b>	<b>6</b>
2.1	<b>Functionality</b>	6
2.2	<b>Security</b>	12
2.3	<b>Usability and Accessibility</b>	15
<b>3</b>	<b><i>Design</i></b>	<b>19</b>
3.1	<b>Design Choices</b>	19
3.1.1	<b>Model-View-Controller Architecture</b>	19
3.1.2	<b>10 Usability Heuristics by Jakob Nielsen's</b>	19
3.2	<b>Design Artifacts</b>	25
3.2.1	<b>User Scenarios</b>	25
3.2.2	<b>Use Case Diagram</b>	26
3.2.3	<b>Sequence Diagram</b>	28
3.2.4	<b>Class Diagram</b>	29
<b>4</b>	<b><i>Product Testing</i></b>	<b>30</b>
4.1	<b>Scenario Testing</b>	30
4.1.1	<b>First scenario</b>	30
4.1.2	<b>Second scenario</b>	33
4.1.3	<b>Third scenario</b>	34
4.2	<b>Integration Testing</b>	36
4.2.1	<b>Success Scenarios</b>	36
4.2.2	<b>Failed Scenarios</b>	38
<b>5</b>	<b><i>Third Sprint Review</i></b>	<b>39</b>
5.1	<b>Sprint Burn-down Chart</b>	39
<b>6</b>	<b><i>Conclusion</i></b>	<b>40</b>

# 1 Application

## 1.1 Product Value

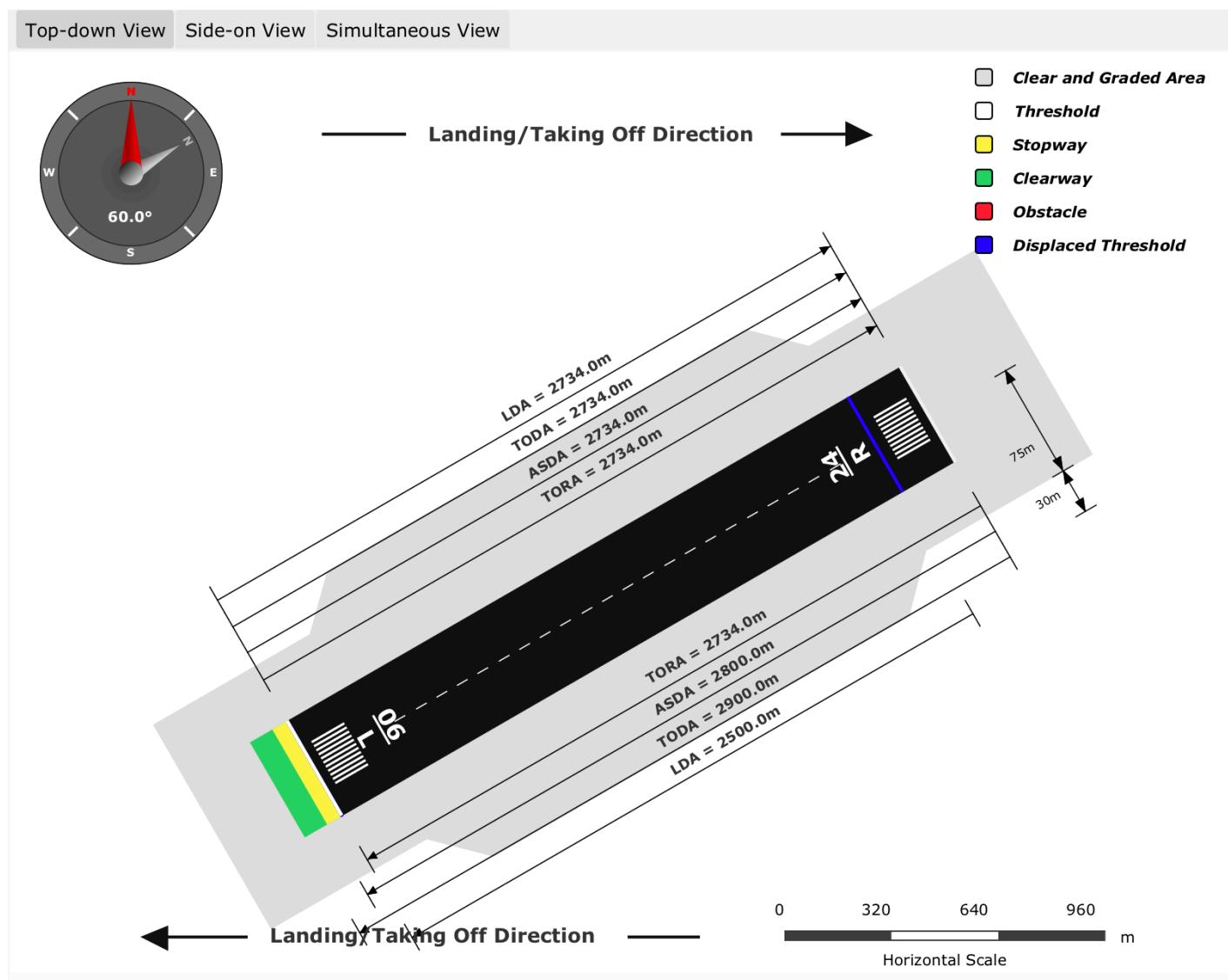
The last code increment of the product has undergone significant improvements to its functionalities based on user and supervisor feedback. The development team has taken into consideration the suggestions and made the necessary changes to enhance the system's usability and functionality. The calculator has been implemented and optimized to provide accurate and reliable results. Additionally, the login feature has been incorporated, allowing access control for users with manager or admin roles to manage users or airports, ensuring the security of sensitive data. The system also allows for the export of regeneration reports or details in the appropriate file formats, enabling users to save and share essential information easily. To cater to individuals with colour blindness, the team has implemented a specialized colour scheme, improving the product's accessibility. These improvements reflect the team's commitment to providing a user-centric design and addressing the needs of a diverse range of users. The product now includes all the necessary functionalities and meets most of the customer's requirements.

## 1.2 Application Details

Following from the previous code increment, additional features have been implemented based on the backlog items we have included for this sprint.

13	As an air traffic controller. I want to automatically rotate the runway strip to match its compass heading. So that the direction of the runway is clear and easily identifiable.	Could
----	---	-------

Upon selection of the runway, the top-down visualisations will automatically rotate the runway so that the direction matches with the compass heading.



7	As an air traffic controller. I want to import and export details of obstacles, airports, and other data using appropriate XML files So that they can be stored/transferred to others.	Should
8	As an air traffic controller. I want to be able to use it at any UK commercial airport. So that I can access it at the control tower.	Should

To incorporate this functionality, we have added a new interface, Airport Manager where users can perform actions such as importing/exporting airport details. They can also edit runway details for selected airport.



Referring to the screenshot above, it can be seen that there's an option to choose an XML files to import a new airport or to export all airport details. XML checker and parser have been implemented to ensure the file content is in the correct format.

After selecting an airport in the table, user can choose to

- Delete the airport
- View its information in textual form on the right-hand side
- Edit the airport details
- Export the airport details in either XML or TXT format

Apart from the airport manager, we have also added a user manager. This is just a page to add new users. For super admin, they can set up manager account for each airport. As for the manager of each airport, they can use this interface to add new users or delete existing users.

The screenshot shows a web application window titled "SEG Runway Project". The main title bar says "MANAGE USER ACCOUNTS". On the right, it says "Logged in as admin" and has a "LOGOUT" button. The central area contains a table with columns: "USERNAME", "NAME", and "AIRPORT". Two rows are visible: one for "manager" (Name: Manager, Airport: Heathrow Airport) and another for "jingru" (Name: Jing Ru, Airport: Dummy Airport). Below the table is a button labeled "Set Up Manager Account", which is highlighted with a yellow rectangle. The top left corner of the window has standard OS X window controls (red, yellow, green).

USERNAME	NAME	AIRPORT
manager	Manager	Heathrow Airport
jingru	Jing Ru	Dummy Airport

Figure 1 Screenshot showing page for super admin to set up new manager account

The screenshot shows a web application window titled "SEG Runway Project". The main title bar says "MANAGE USER ACCOUNTS". On the right, it says "Logged in as manager" and has a "LOGOUT" button. The central area contains a table with columns: "USERNAME", "NAME", and "AIRPORT". One row is visible: "atc" (Name: Air Traffic Controller, Airport: Heathrow Airport). Below the table are two buttons: "Set Up User Account" and "Delete Selected User Account", which are highlighted with a yellow rectangle. The top left corner of the window has standard OS X window controls (red, yellow, green).

USERNAME	NAME	AIRPORT
atc	Air Traffic Controller	Heathrow Airport

Figure 2 Screenshot showing page for manager to set up new user account

16	As an air traffic controller. I want to print out the situation of the runway in textual format. So that it could be revise with ease.	Won't
----	--	-------

This have been implemented as an option to print out a redeclaration report after performing calculation. Format of the report can be seen in the next section.

15	As an air traffic controller. I want to zoom and pan the views. So that I can focus on the part of the runway that I am working with.	Won't
----	---	-------

This has also been implemented in this code increment where user can zoom in or out of the view by pressing command key and scrolling or drag across the display by pressing the command key and drag in different direction.

# 2 Extensions

## 2.1 Functionality

### 1. Zoom and pan the views (add label rotating)

The system allows the user to perform some actions such as zooming in and out of the visualisations, dragging the visualisation to focus on different areas. Besides that, the user can also rotate the views, with the compass heading automatically updated based on the rotation angle.

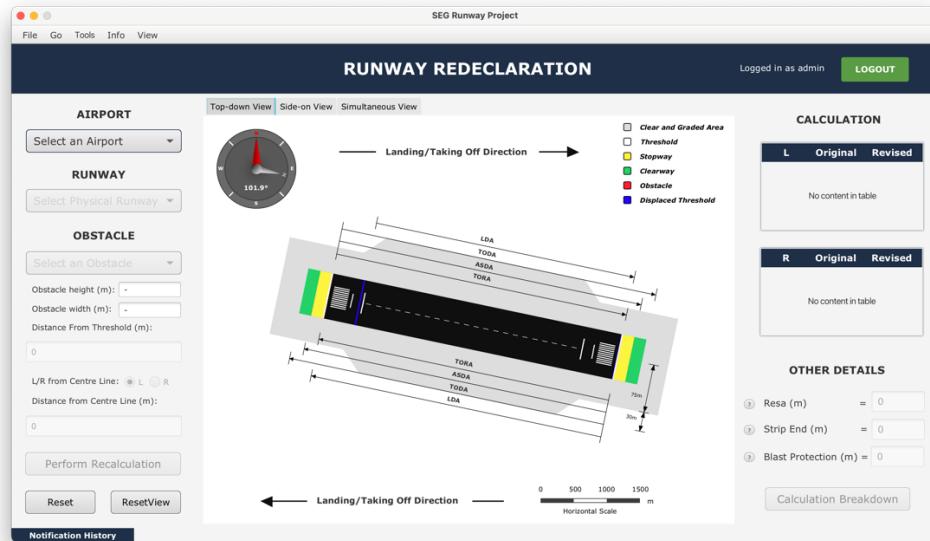


Figure 3 Diagram showing views being rotated

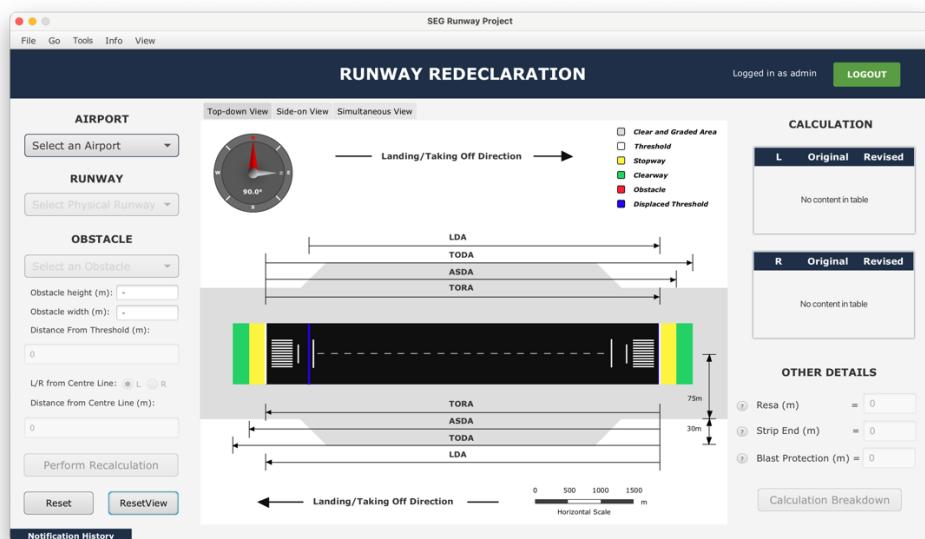


Figure 4 Diagram showing views at normal state without rotation or dragging

There are also some usability features in this functionality where the label will be rotated accordingly so that they are not upside down when the user rotates them by 180 degrees.

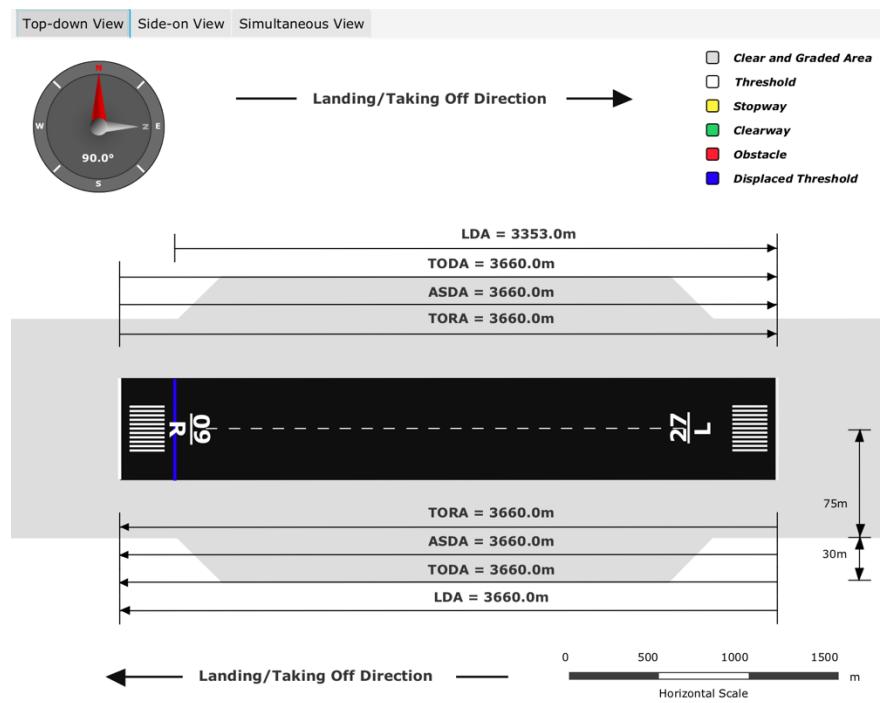


Figure 5 Diagram showing the original orientation of the top-down visualisation

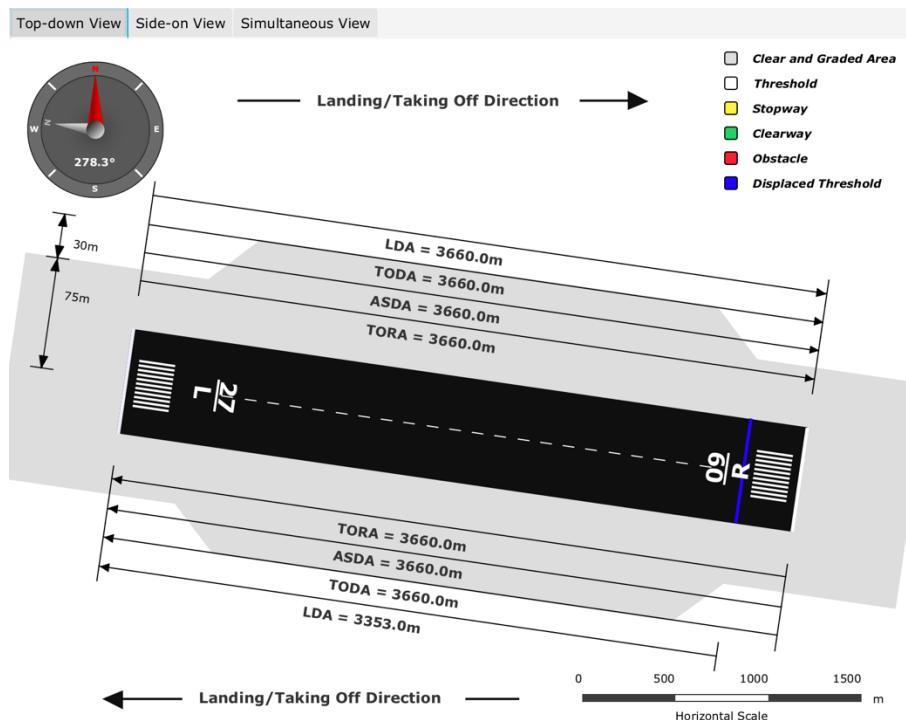


Figure 6 Diagram showing label being rotated automatically when rotation > 180 degrees

## 2. Export displays in different formats

There is an option for the user to export the visualisations, including Side-On, Top-Down and Simultaneous View as JPEG or PNG images.

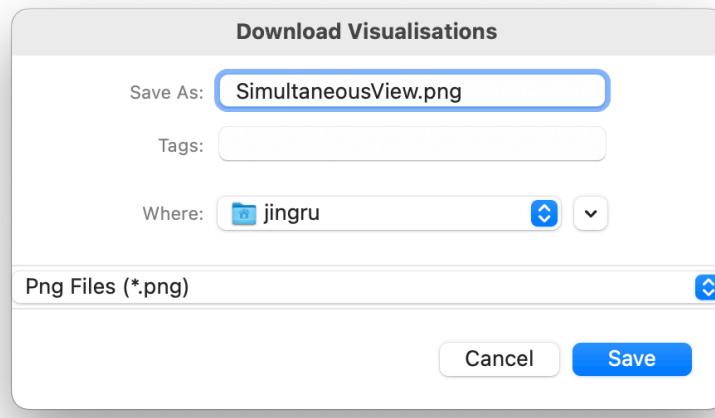


Figure 7 User can select a directory to save the display

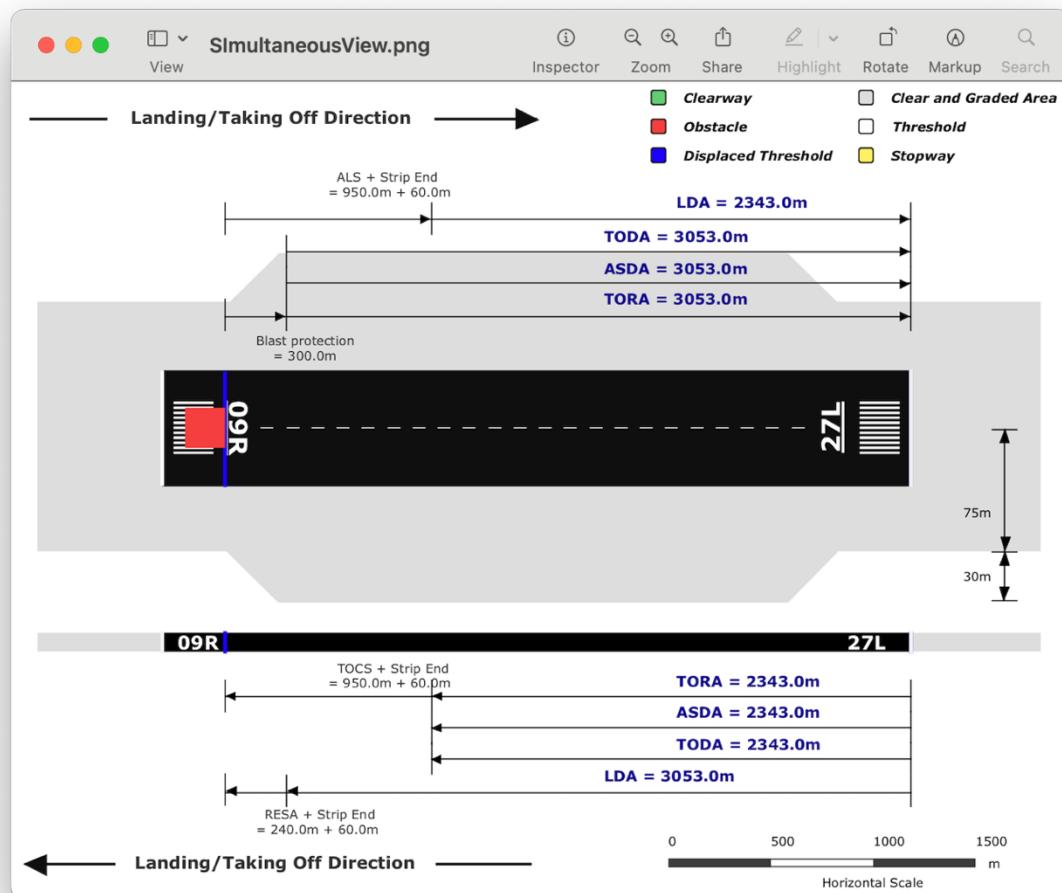


Figure 8 Diagram showing an example of an image being exported

### 3. Generate Redeclaration Report

(Code extracted and modified from: <https://github.com/cekpowell/runway-redeclaration-tool/tree/main>)

After performing the calculation, the user is given the option to print out the information as a report in PDF. The report will include all the necessary information for the calculation.

The first page of the report will be the details of the airport and obstacle and a table showing a comparison between original runway parameters and revised parameter values.

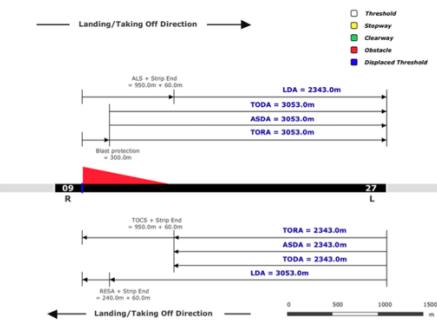
The second page of the report will be showing the calculation breakdown for the runway in both directions so that it can be compared with manual calculations.

<b>Calculation report</b>				
Generated by Admin on 2023-04-23 21:15:49				
<b>Airport details</b>				
Name: Heathrow Airport Runway: 09R/27L Runway length: 3660.0m Blast protection: 300.0m Strip end: 60.0m				
<b>Obstacle details</b>				
Name: Boeing 747 Width: 69.0m Height: 19.0m Distance from threshold: 0.0m Distance from center: 0.0m from left				
<b>Re-declaration breakdown</b>				
<b>09R Take-Off Away Landing Over</b>				
TORA = Original TORA - Blast Protection - Distance from Threshold - Displaced Threshold = 3660.0 - 300.0 - 0.0 - 307.0 = 3053.0				
ASDA = (R) TORA + STOPWAY = 3053.0 + 0.0 = 3053.0				
TODA = (R) TORA + CLEARWAY = 3053.0 + 0.0 = 3053.0				
LDA = Original LDA - Distance from threshold - Strip End - Slope Calculation = 3353.0 - 0.0 - 60.0 - 19.0*50.0 = 2343.0				
<b>27L Take-Off Away Landing Over</b>				
TORA = Original TORA - Blast Protection - Distance from Threshold - Displaced Threshold = 3660.0 - 300.0 - 0.0 - 0.0 = 2343.0				
ASDA = (R) TORA + STOPWAY = 2343.0 + 0.0 = 2343.0				
TODA = (R) TORA + CLEARWAY = 2343.0 + 0.0 = 2343.0				
LDA = Original LDA - Distance from threshold - Strip End - Slope Calculation = 3660.0 - 0.0 - 60.0 - 19.0*50.0 = 3053.0				
<b>Table of original and revised values</b>				
Runway designator	TORA	TODA	ASDA	LDA
Original Values				
09R	3660.0m	3660.0m	3660.0m	3353.0m
27L	3660.0m	3660.0m	3660.0m	3660.0m
Recalculated Values				
09R	3053.0m	3053.0m	3053.0m	2343.0m
27L	2343.0m	2343.0m	2343.0m	3053.0m

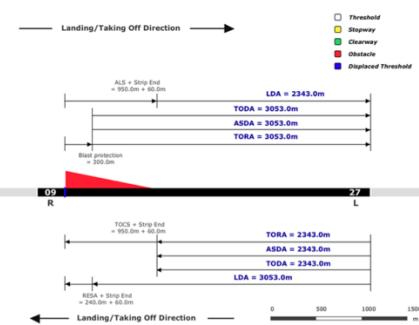
The last three pages will be the snapshots of runway visualisations.

#### Visualisations

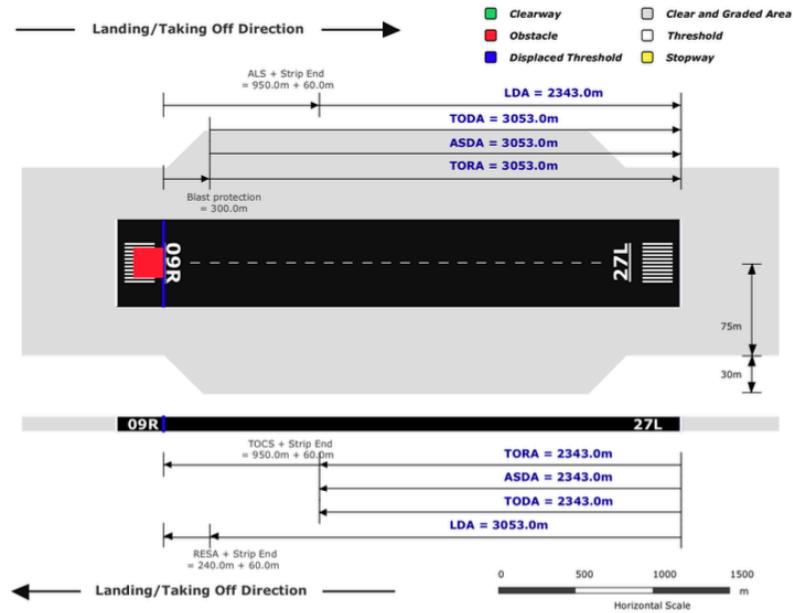
##### Side-On Visualisation



##### Top-Down Visualisation



#### Simultaneous Visualisation



## 4. Dark Mode and Light Mode Theme

As part of the extension, the team has implemented an option for user to switch between dark and light mode for the interface. We keep the same visualisation display in both themes because we feel that the visualisation will be clearer with lighter background instead of changing it to darker ones to suit the theme.

However, as future work, further redesigning of the visualisations could be carried out to improve on the visualisations for dark theme.

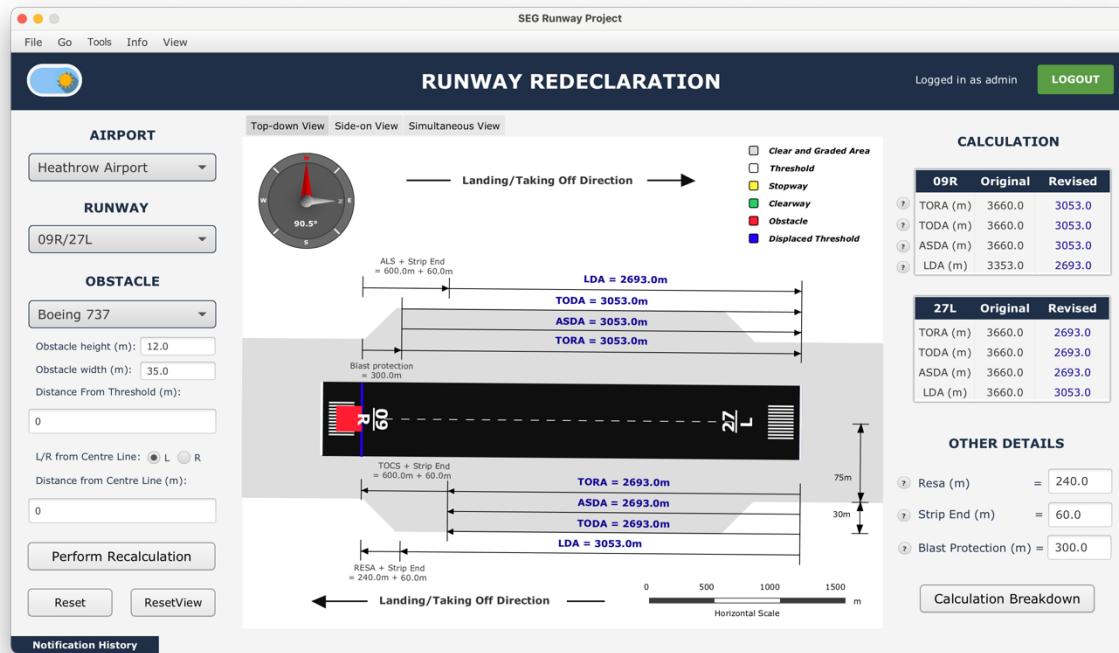


Figure 9 Screenshot showing system in light mode theme

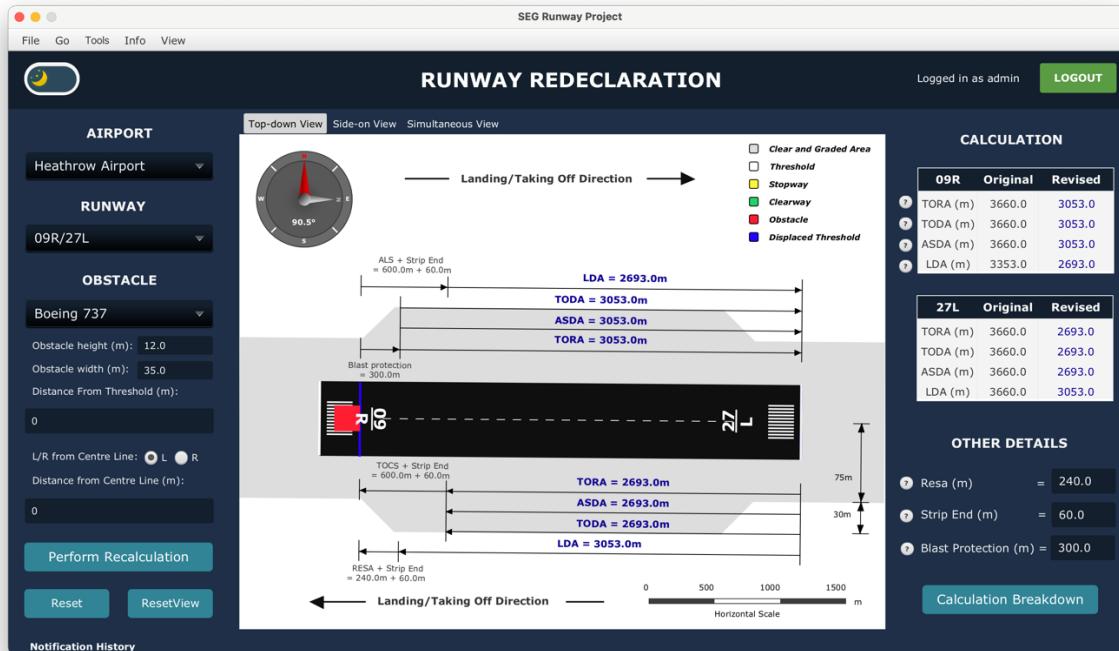


Figure 10 Screenshot showing system in dark mode theme

## 2.2 Security

To protect the system's security so that important information is not tampered with by unauthorised users, we have implemented some security features including a login page, access control system and automatic logout.

To access the interface, users are required to log in to the system. To ensure the security of user credentials, passwords are encrypted before being stored. Currently, these encrypted passwords are saved in XML files. However, in real-life cases involving a larger user database, it is recommended to set up an SQL server to store user information, which can further enhance the security and protection of the data. The encryption of passwords is essential in protecting user accounts from unauthorized access. If a password is not encrypted, anyone with access to the database can easily retrieve it and gain access to the user's account. Therefore, encryption provides an added layer of security by converting the password into a non-readable format, which can only be deciphered with the correct decryption key.

### 1. Login features

There is only a login option but not the option to sign up or create a new account is absent in our implementation. This is because we have designed the system such that accounts are being set up by admin or managers as part of the access control procedures. This will be further explained in the next part. Validation of credentials will be carried out on user input by checking with data stored in the database.

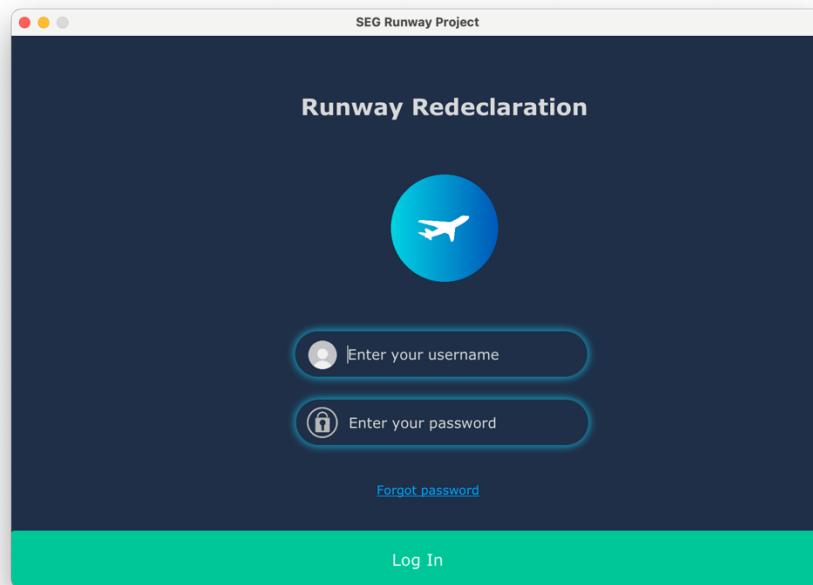


Figure 11 Diagram showing the login window

During the regular meetings, our supervisor and user have raised the concern of managers and super admin forgetting their passwords and they feel like there should be an option to reset passwords like any other system. In the real situation, the procedure will involve users being directed to a link or getting an email with links to reset the password. In our case, we have simply created a dummy pop-up page showing a window for the user to re-enter a new password. We did not follow through with the actual process fully as we think that it would not be feasible or necessary for this coursework.

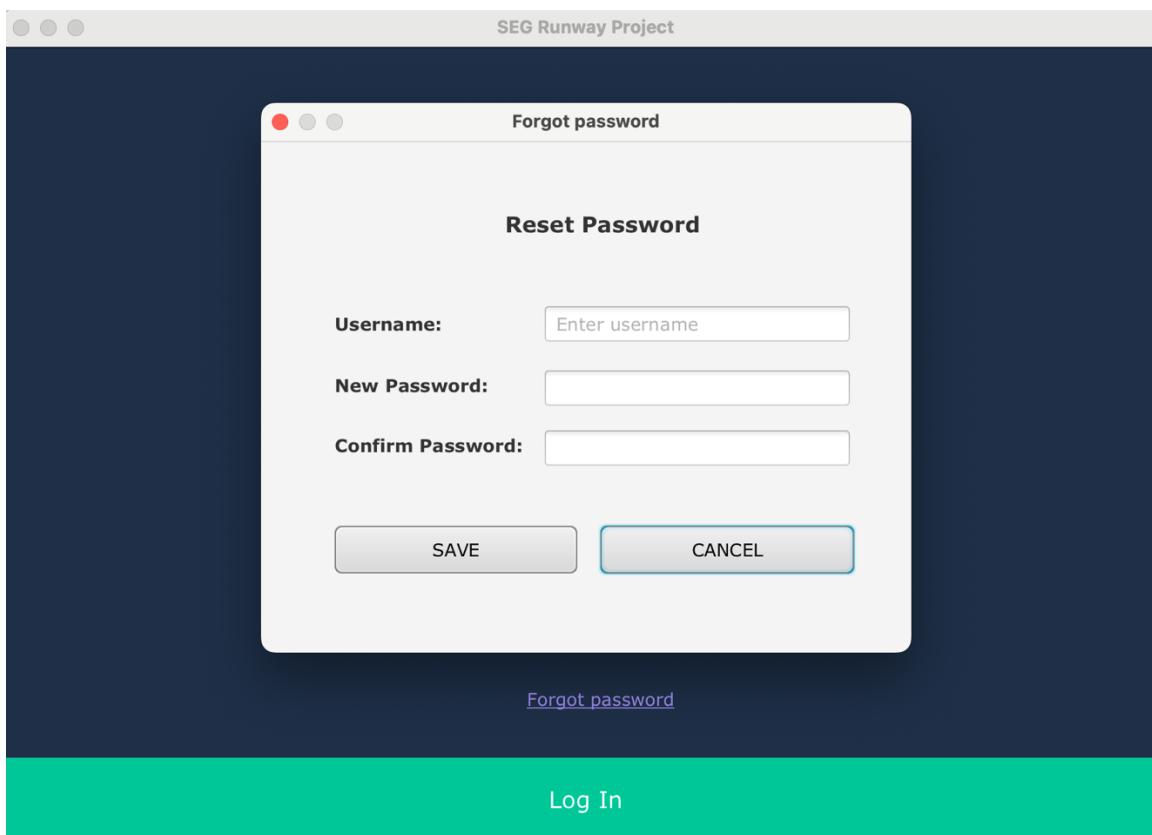


Figure 12 Diagram showing pop-up when the user needs to reset password

In our implementation, when user clicks on the ‘Forgot password’ text on the login page, they will be shown a window as shown above, and then they will have to fill in their username and key in the new password twice. Password validation is implemented where the password chosen must be at least 6 characters long and must contain at least one number, one special character and a mix of uppercase and lowercase letters.

## 2. Access Control System

As part of the security consideration, we have set up an access control system where we assign roles to users and different types of users will have different levels of access to the system.

We are using the role-based access control (RBAC) policy where access to parts of the system is either granted or denied based on predefined responsibilities and rules for a specific role.

**Table 1 Access Control:** List of access rights for each role in the system

Roles	Access Rights
Super Admin	<p>Super admin will have access to the full functionalities of the system including the redeclaration calculator.</p> <p>They are also responsible for:</p> <ul style="list-style-type: none"> <li>• Managing airports</li> </ul> <p>This includes adding/importing airports in XML formats, deleting airports or editing airport details.</p> <ul style="list-style-type: none"> <li>• Managing managers</li> </ul> <p>They are in charge of setting up manager accounts for an airport after an airport has been added.</p>
Manager	The manager can access the calculator and is also in charge of adding new users (air traffic controllers) to the system. The users are automatically added under the airport which the manager is associated with. The manager of each airport will not have access to other airport information or runways.
Air traffic controller	They are normal users of the system and can only access the redeclaration calculator.

### 3. Automatic logout after inactivity

This is a feature suggested by the supervisor and user. Based on their recommendation, we have set an inactivity timer. The timer keeps track of user activity and if the system is idle (no actions performed) after some time, the user will be prompted to log out of the system. They are also given the option to continue working if they do not wish to proceed to log out. We have set the timer to 3 minutes as default, but it is customisable based on different rules and regulations.

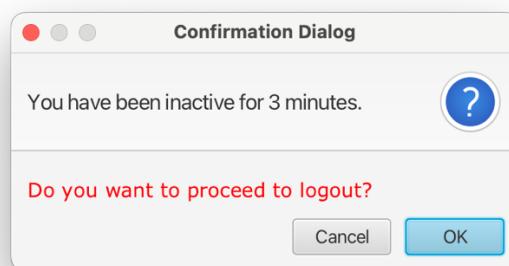


Figure 13 Screenshot showing pop-up prompting user to logout after inactivity

## Future work

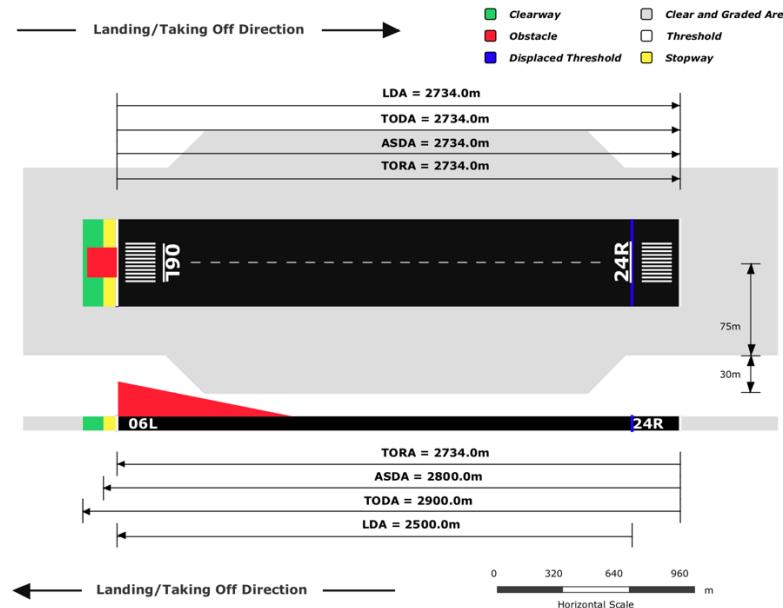
Some of the extensions to improve the security features are not feasible for the amount of time in this coursework, but to secure the system in real cases, we can also implement multi-factor authentication, where users are required to provide additional verification, such as a code sent to their phone, before accessing their account.

Lastly, we could also use secure communication protocols such as HTTPS to encrypt the transmission of login credentials between the user and the server. This prevents attackers from intercepting the login information and gaining access to the user's account.

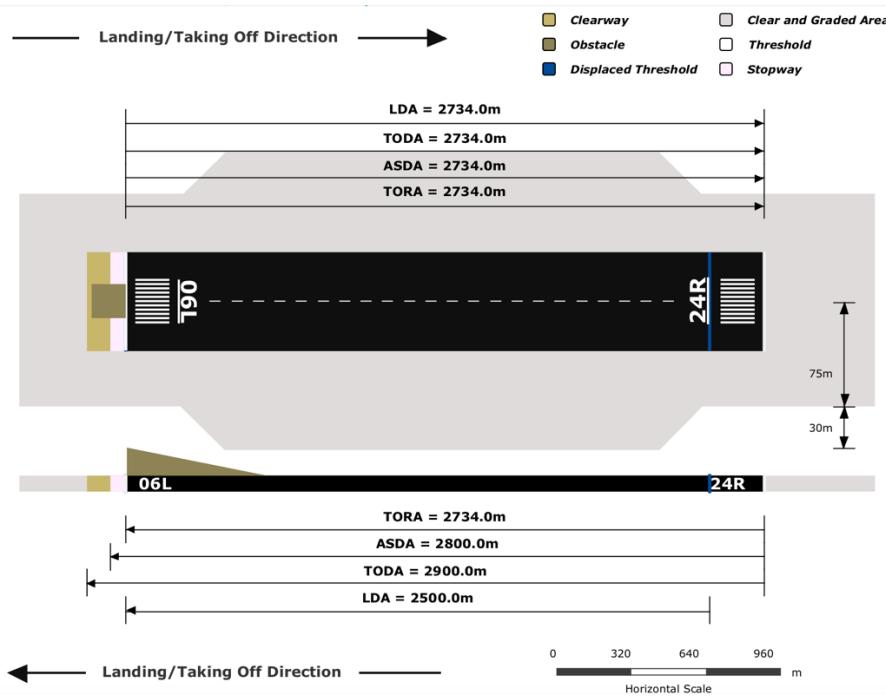
## 2.3 Usability and Accessibility

To improve on usability and accessibility of the system, we have considered one of the extensions suggested in the project definition, which is to include various themes for colour-blind users. Using a free simulator tool online, we were able to simulate the visualisations and view them from colour-blind users' viewpoints. We have selected distinct colours for visualisation purposes so that each component could be easily perceived and differentiated by users. However, we realised that the visualisations look completely different for different types of colour-blind users, mainly for people with red-green and blue-yellow colour blindness. This causes some of the colours to be undistinguishable, making it very difficult for them to identify even if they refer to the legend provided.

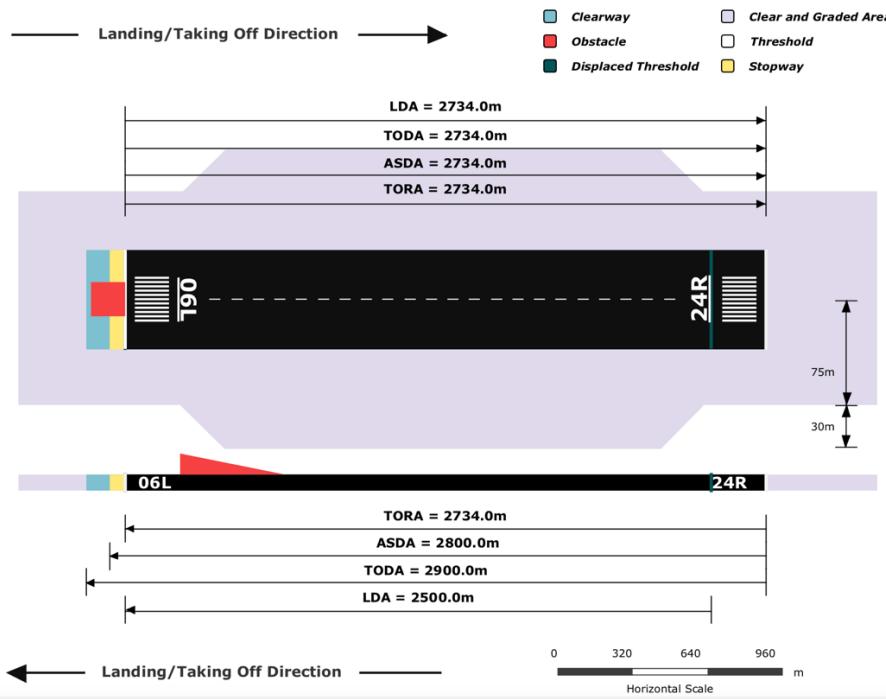
The first picture below shows the visualisation viewed by a user without colour blindness, it can be seen that all the colours used are very distinct and could be easily identified.



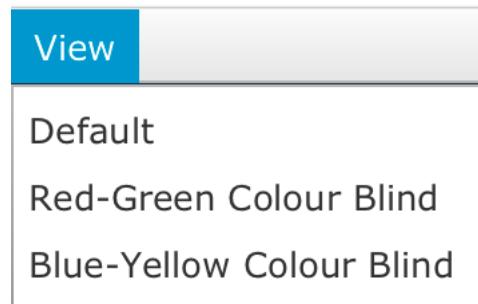
For a user with red-green colour blind, below is a simulation of visualisation perceived by them. Take note of the low contrast between the clearway and obstacle, between threshold and stop way and also between the runway and displaced threshold.



A simulation of visualisation perceived by a user with blue-yellow colour blindness. Even though most of the colours can be distinguished, some lines such as the displaced threshold are not clear as compared to the original visualisation.

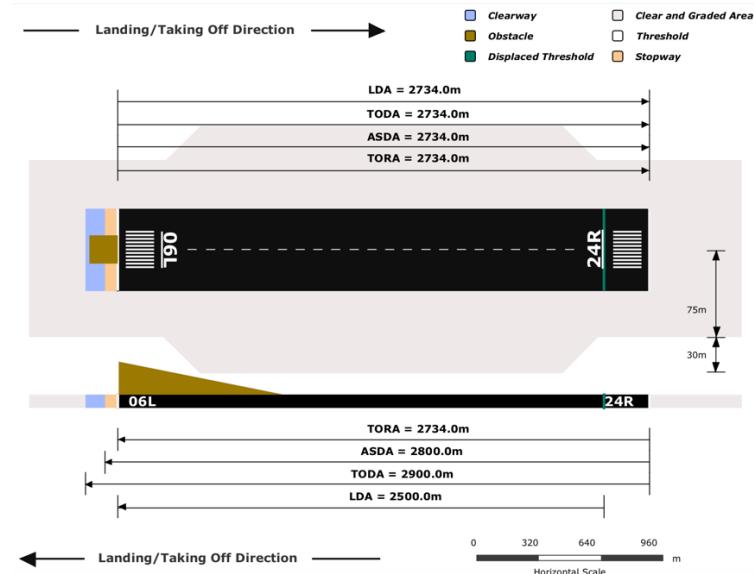
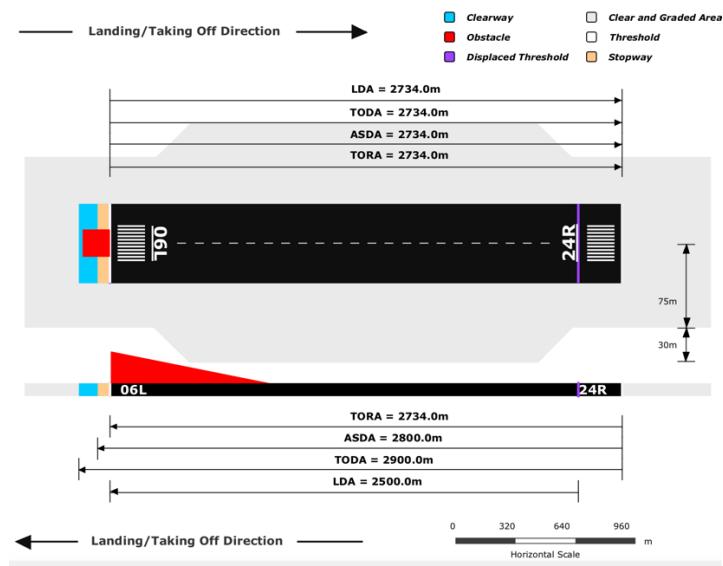


As a solution, there is an option in the menu bar to switch between different types of views to optimise the experience for users with special needs.



## 1. Red-Green Colour Blind

The first image shows visualisation perceived by users without colour blindness and the latter shows visualisation perceived by users with red-green colour blindness.



## 2. Blue-Yellow Colour Blind

The first image shows visualisation perceived by users without colour blindness and the latter shows visualisation perceived by users with blue-yellow colour blindness.

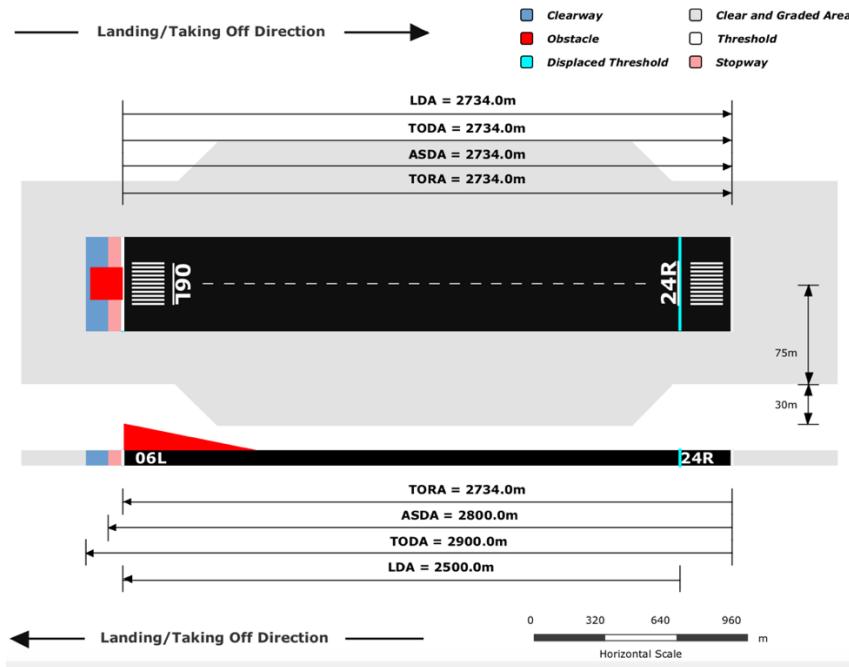


Figure 14 Screenshot showing what a user without colour blindness sees

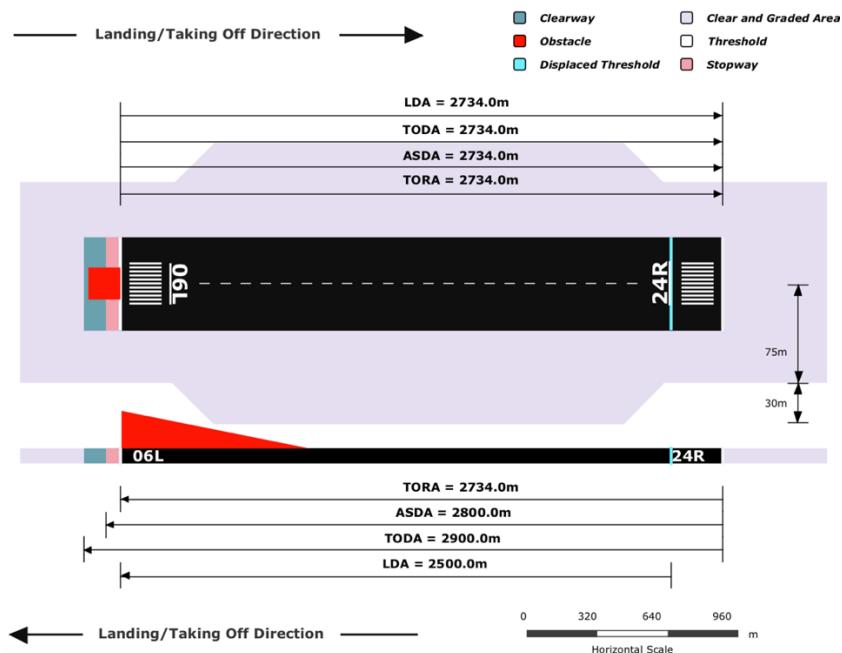


Figure 15 Screenshot showing what users with Blue-Yellow colour blindness see

# 3 Design

## 3.1 Design Choices

### 3.1.1 Model-View-Controller Architecture

For this increment, we are still utilising the Model-View-Controller architecture and modifying the structure slightly to enable visualisations to be included.

### 3.1.2 10 Usability Heuristics by Jakob Nielsen's

In this increment, since it will be the last code increment, we have decided to analyse the system design based on the 10 usability heuristics, which are rules of thumbs for interaction design.

#### 1. Visibility of system status

To keep users informed about what is going on, the system provides feedback such as notifications after important actions such as performing calculations and importing airports. Users will be notified if their actions have been successful or failed so that they can determine their next steps.

The feedback is provided in two forms, notification banner in the main interface or notification pop-ups that will fade after some time.

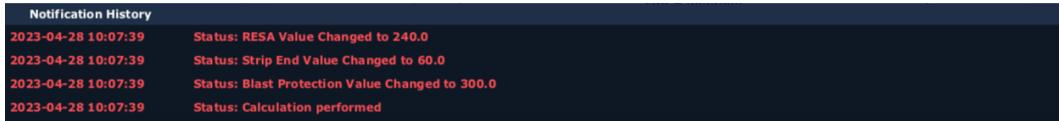


Figure 16 Screenshot showing notification histories when the user expands the notification banner

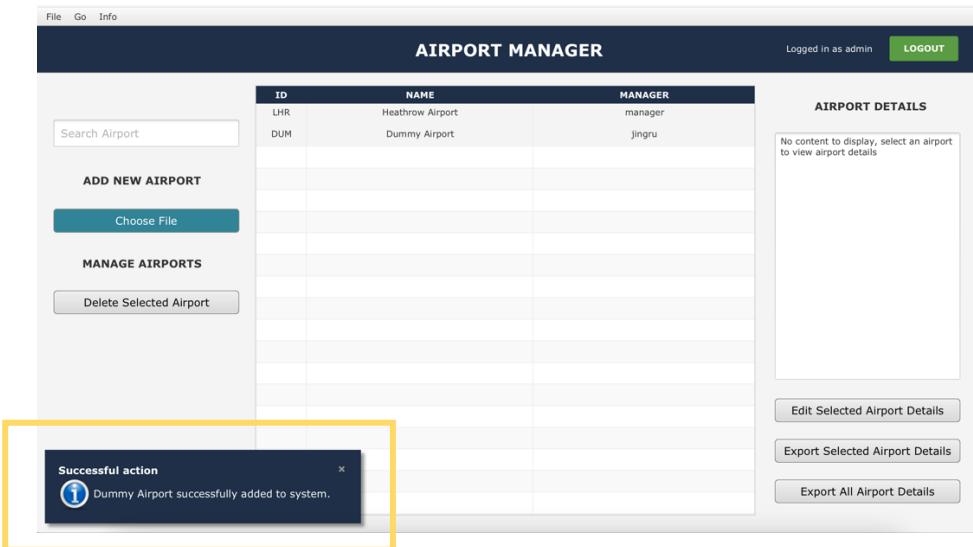
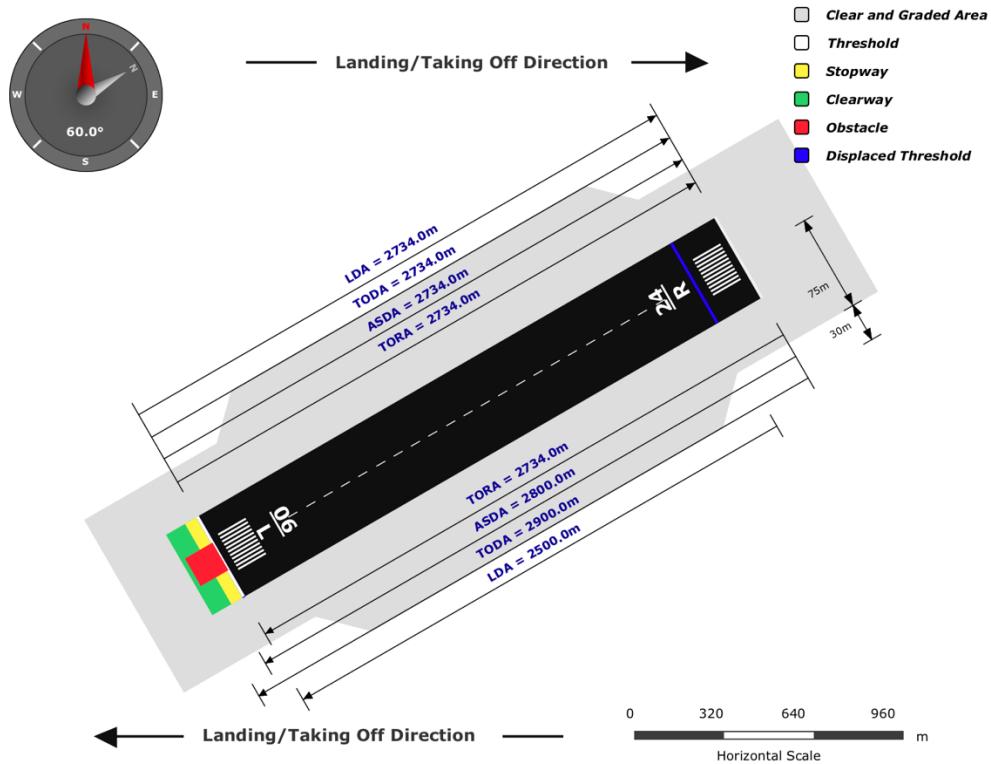


Figure 17 Screenshot showing notification pop-up when the user successfully added a new airport

## 2. Match between system and real world

One of the features of the system that match the system with the real world is the rotation of the runway to match the compass heading. This can allow users to visualise the actual angle of the runway in the real world.



In the screenshot above, runway 06L/24R is being rotated to match the actual orientation of the runway. 06 means that the runway is heading towards 60° whereas 24R is facing the 240° direction.

## 3. User control and freedom

There are options to cancel actions in case users accidentally perform the actions or wish to exit the current process such as importing airport or editing details.

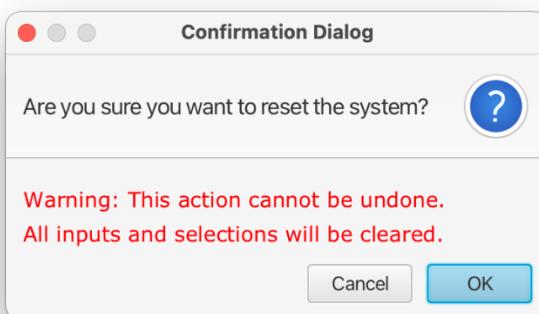


Figure 18 An example of cancel option in confirmation dialog for a user action

The screenshot below shows the window when the user wants to edit airport details, in the case that they wish to discard changes and return to the interface, they can click on the ‘cancel’ button, where all the previous changes will be discarded and reverted.

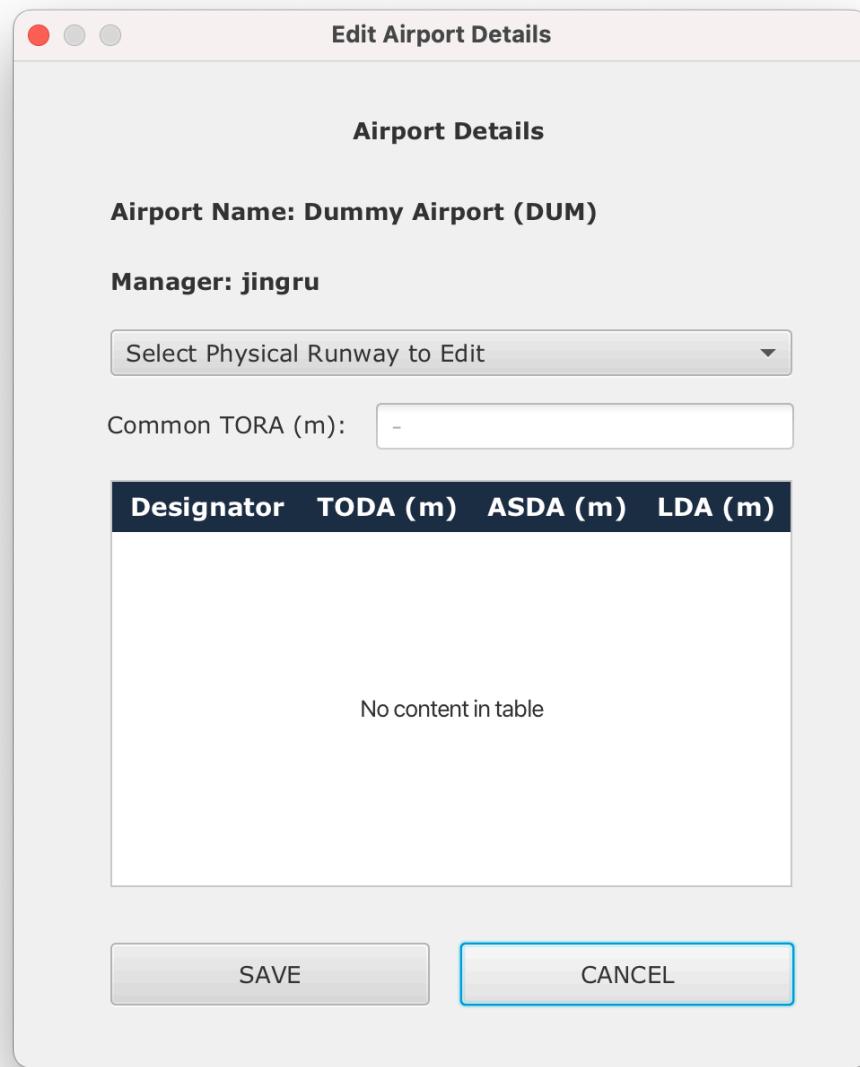


Figure 19 Screenshot showing cancel option when editing the airport details

#### 4. Consistency and standards

Consistencies are maintained throughout the system by standardising similar actions. For example, in both the airport manager and user manager, to delete or edit, the user needs to select a specific row.

#### 5. Error prevention

Apart from error messages, error prevention is being implemented to prevent problems from occurring in the first place. Firstly, we have confirmation dialogues when the user tries to perform dangerous actions such as deleting the airport or resetting the system.

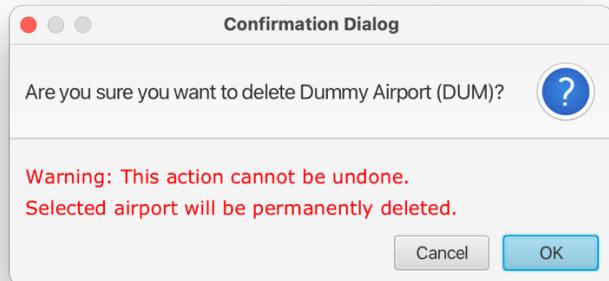


Figure 20 An example of a confirmation dialog as error prevention

Other than that, when handling user input in the main interface, we have disabled buttons or fields so that users have to key in the required information before proceeding to other selections or inputs.

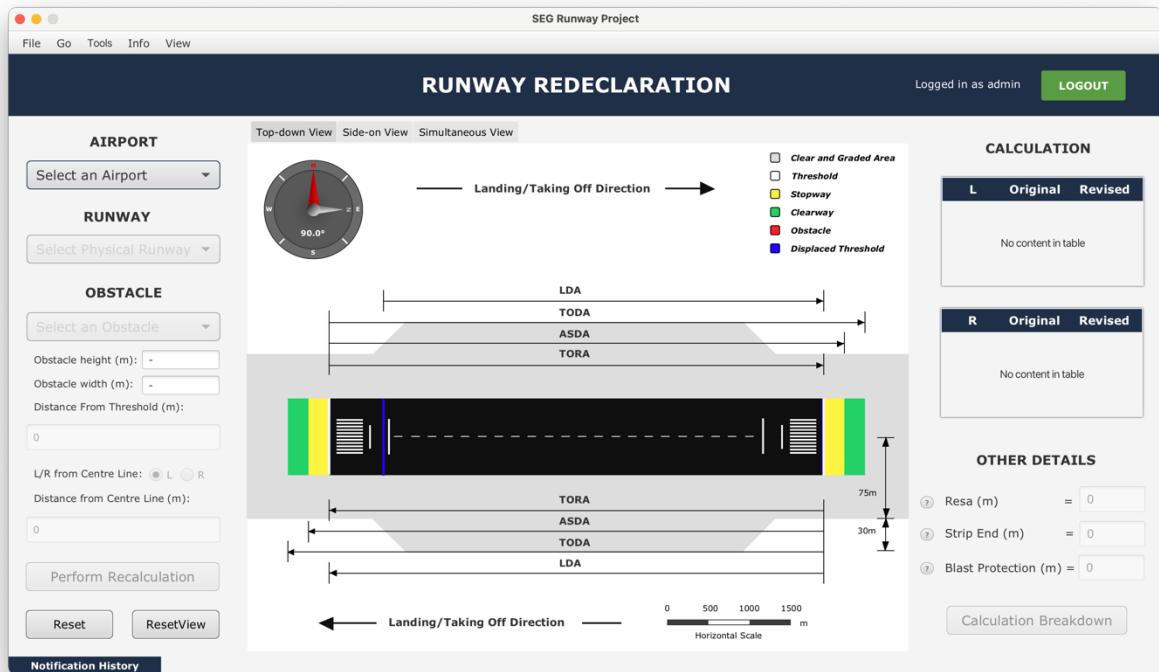


Figure 21 Screenshot showing some buttons being disabled before users select an airport

## 6. Recognition rather than recall

To minimise the user's memory load, some information is made easily retrievable when needed. For example, we have a small info button, when hovered on will display a definition or short explanation of the label.

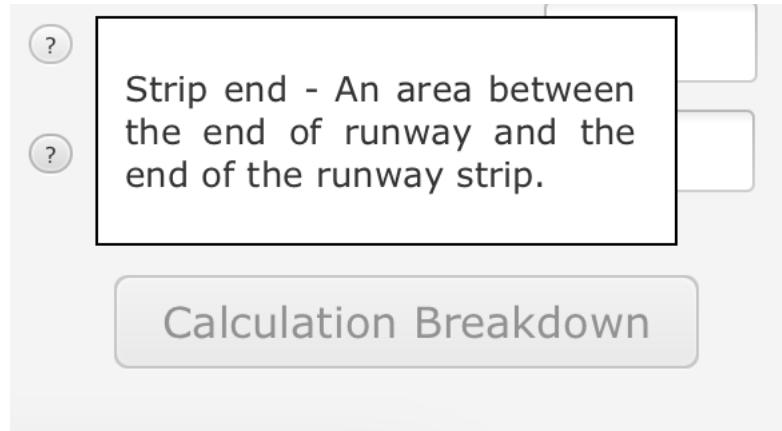


Figure 22 Screenshot showing information displayed when user hovers on info button besides Strip End

## 7. Flexibility and efficiency of use

The system provides flexibility by adding shortcuts for actions, the shortcuts are hidden from novice users in the menu bar. This can speed up the interaction for an expert user so that the design caters for both inexperienced and experienced users.

For example, when interacting with visualisations, novice users can rotate the display by pressing the shift button and dragging on the display or by clicking on the menu item. At the same time, we have shortcuts for the action where users could perform the same action of rotating by using the specific shortcuts. When switching between different colour themes, novice users may need to navigate the menu bar, by first clicking on the View item, and then selecting the corresponding theme, however, once familiar with the system, expert users can simply click on F1/F2/F3 buttons accordingly.

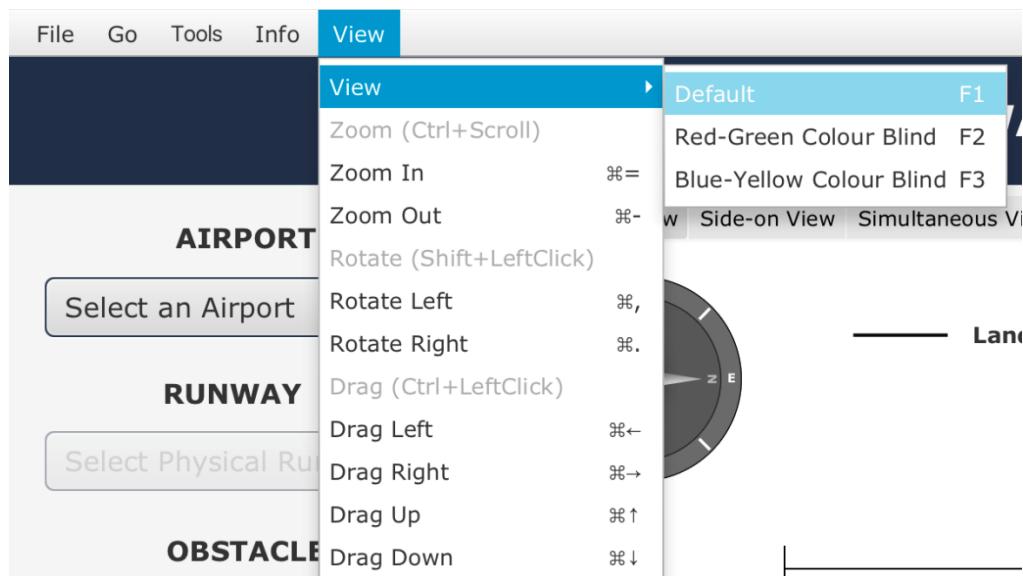
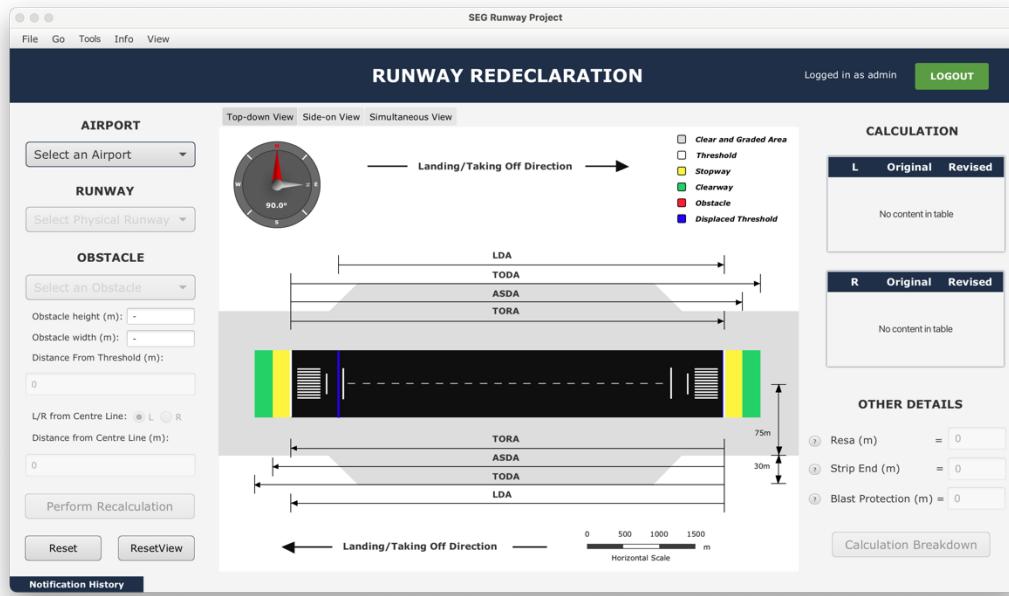


Figure 23 Screenshot showing shortcuts besides functions in menu bar

## 8. Aesthetic and minimalist design

The design team has made an effort to implement a minimalist and aesthetic design. This includes deciding on simple yet professional colour theme and also on reducing or hiding irrelevant information from the interface.



As seen from the screenshot of the main interface, the design has been made simplistic where we explicitly hide redundant buttons or options either in the menu bar or expandable banner. For example, options like exporting or switching themes are hidden in the menu bar, whereas notification history are shown as a single line banner at the bottom part of the interface showing the most recent change but upon clicking, there is an option to expand the banner to show the entire notification history.

## 9. Help users recognise, diagnose and recover from errors

Throughout the interface, whenever there is input error or failed actions, error messages will be shown, explaining the error in simple language, and providing a hint whenever necessary so that users could retry and recover from the specific error.

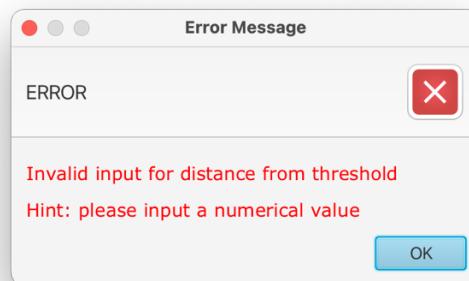


Figure 24 Screenshot shows an example of an error message for invalid input

## 10. Help and documentation

We have also provided help and documentation such as a downloadable user guide. Feel free to refer to the user guide attached in the submission folder for more information.



Figure 25 Some documentations and information provided

## 3.2 Design Artifacts

In this increment, we are still using Unified Modelling Language (UML), a modelling tool that was used for planning and designing the general structure of our application. We decided to include a use case diagram, sequence diagram, class diagram, and storyboard as key design artefacts alongside a user scenario.

### 3.2.1 User Scenarios

User scenarios were introduced during the development phase so that our team would have a better understanding of how users interact with the system.

#### 3.2.1.1 Scenario 1

During peak hour, an airplane is trying to take off while another airplane was blocking the runway. Due to the obstruction, the air traffic controller responsible for the runway wants to have a quick redeclaration of the runway to see if the actual process is worthwhile. He also wanted to have a breakdown calculation to see if the calculations were correct.

#### 3.2.1.2 Scenario 2

A new air traffic controller just joined the crew, and the manager of the airport wants to set up a new runway redeclaration system account for the new crew.

#### 3.2.1.3 Scenario 3

The system admin is trying to import new airport to the system through an XML file containing all the airport information. After importing the airport, he found out from the airport details section that there is an error in the airport, he wants to edit the airport.

### 3.2.2 Use Case Diagram

This use case diagram was built based on the scenario above. The main focus of this diagram is to visualise how air traffic controllers interact with the system and help the user to understand what the development team will come up with in the early stage of development to clear up any misunderstanding between them.

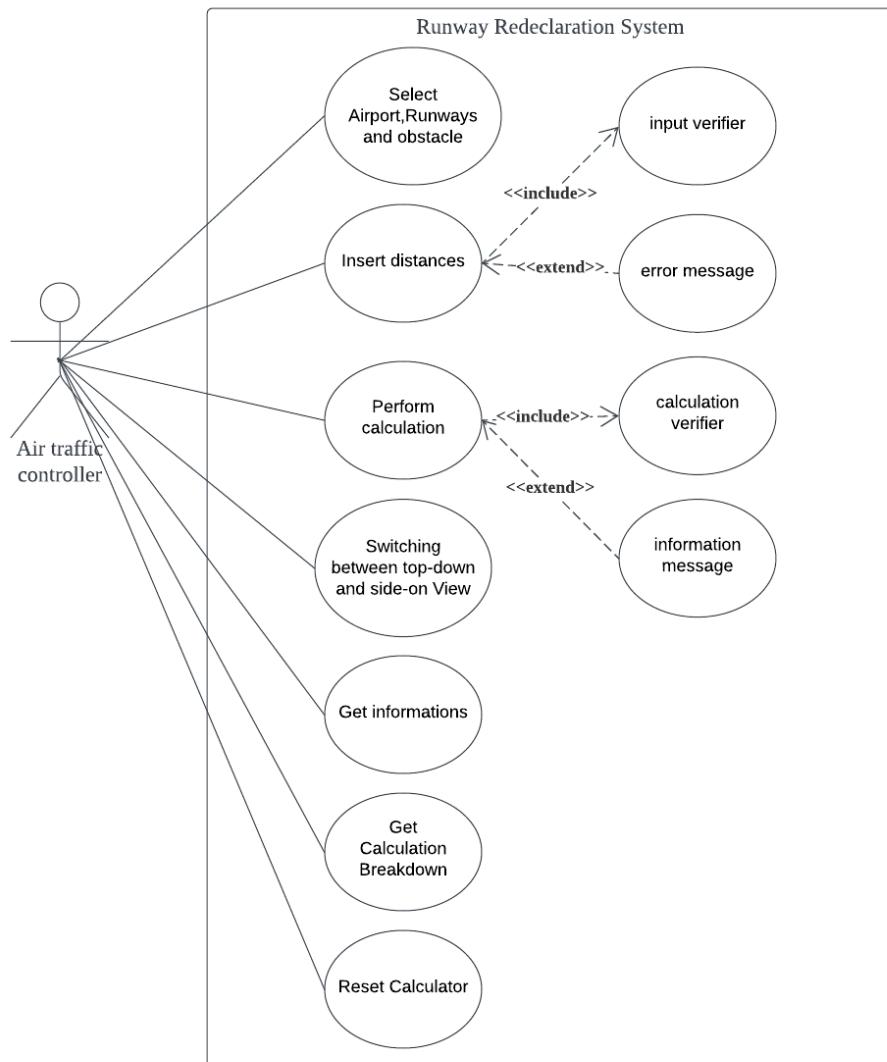


Figure 26 Use case diagram of scenario 1

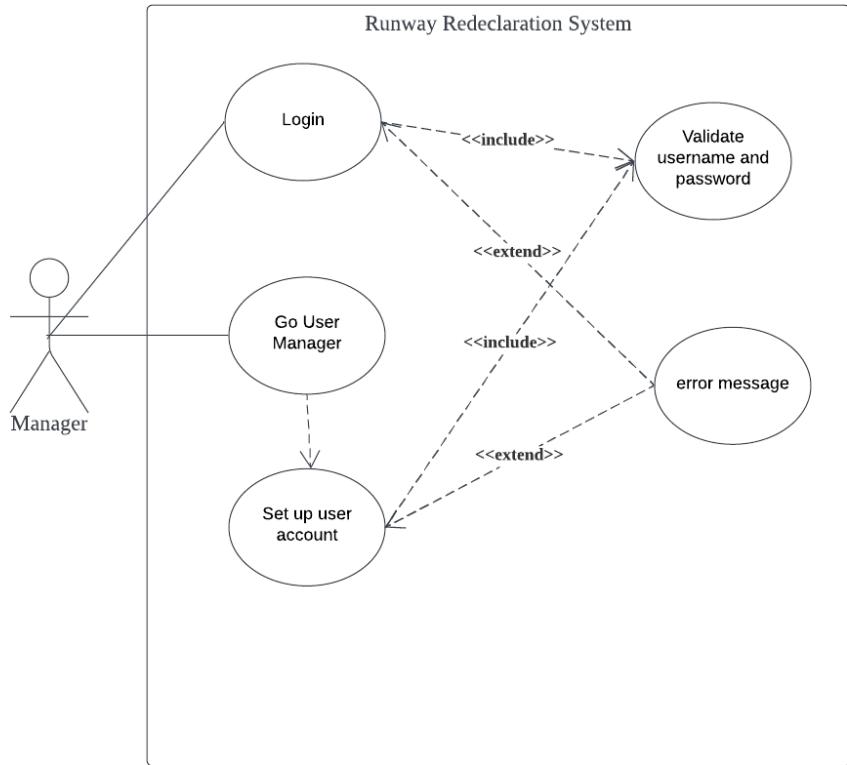


Figure 27 Use case diagram of scenario 2

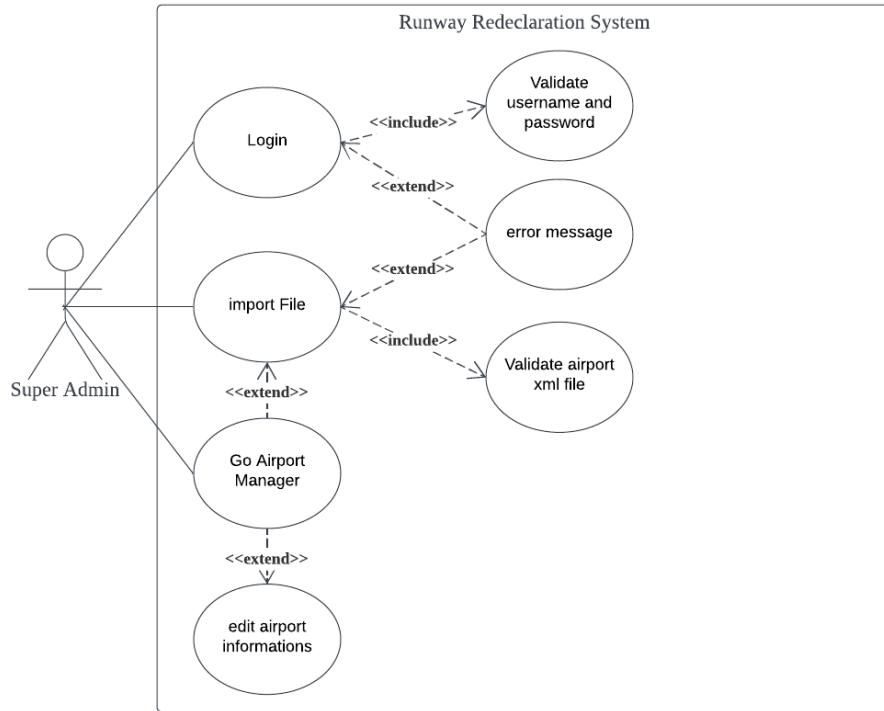


Figure 28 Use case diagram of scenario 3

### 3.2.3 Sequence Diagram

This sequence diagram illustrates how the actor (air traffic controller) interacts with a different part of the system that was based on MVC architecture. It also shows the flow of interactions between systems when a task is performed by the user. In this scenario, the user is trying to redeclare runway declaration and access the calculation breakdown feature to compare his paperwork to the system.

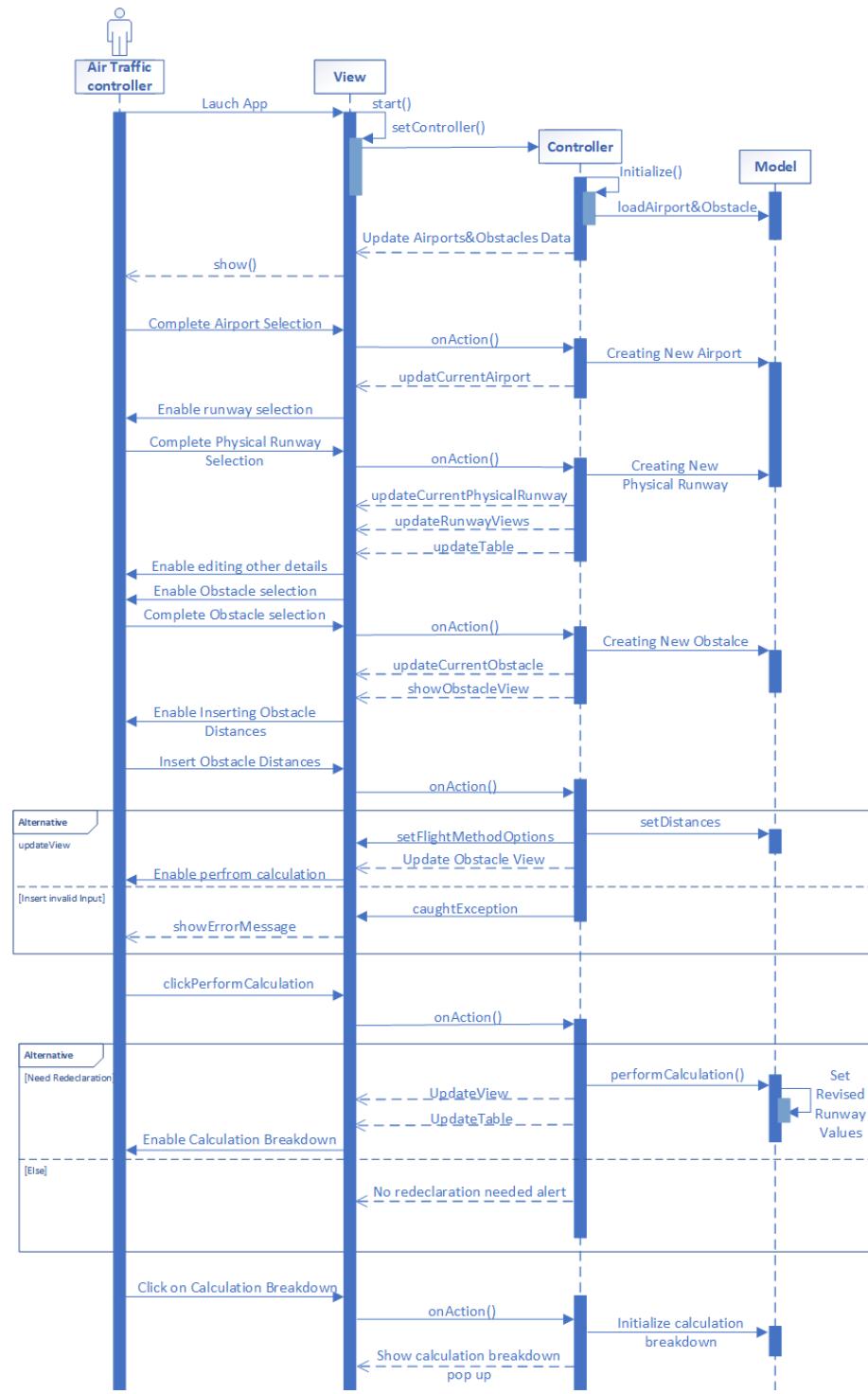
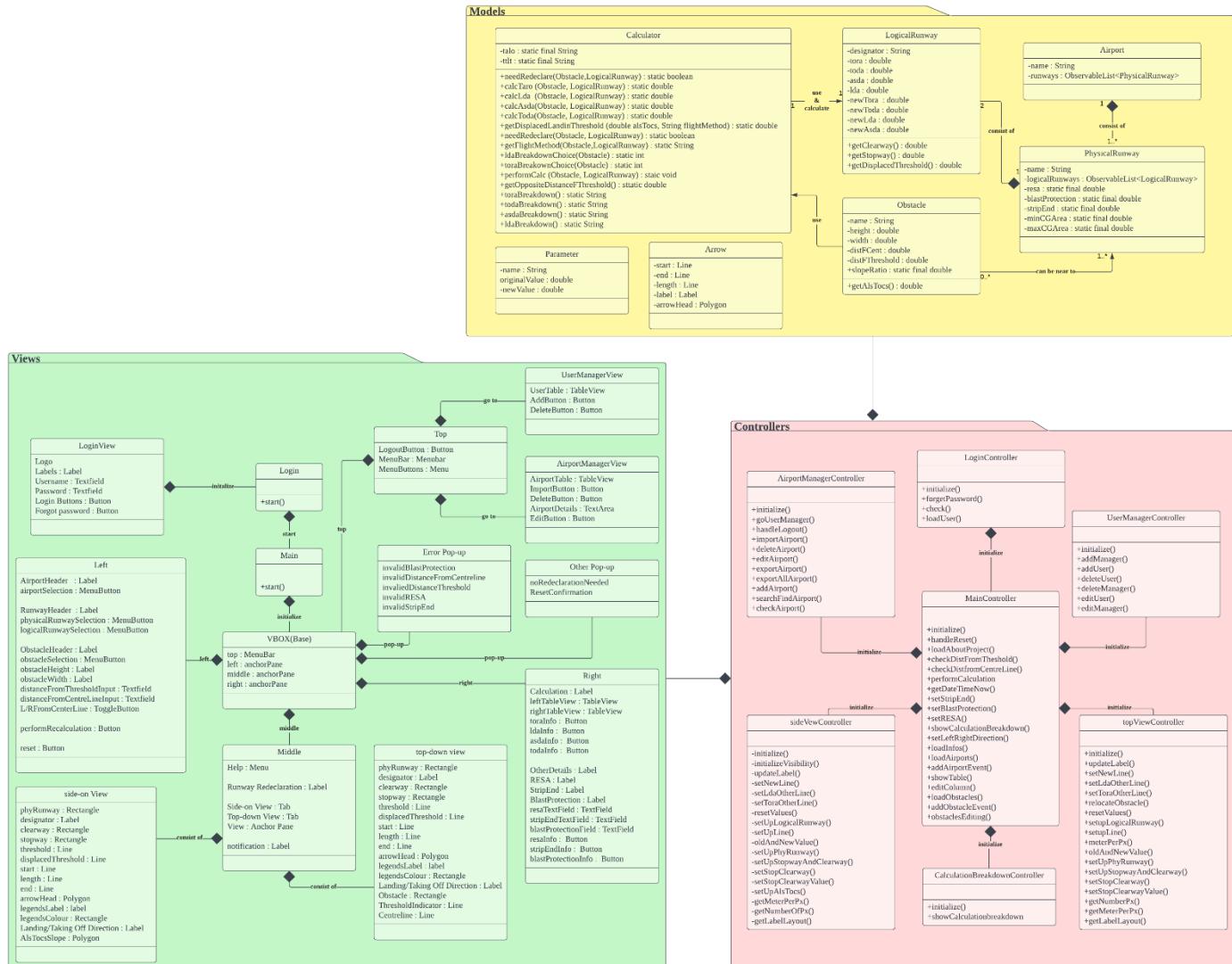


Figure 29 Sequence diagram

### 3.2.4 Class Diagram

A class diagram was created to provide a clear understanding of the system structure among all the team members. This allows us to further divide our tasks in more detail while also working towards the same goal even when we are not physically working together all the time.

We are using the Model View Controller framework (MVC) to design the application. This class diagram consists of 3 different packages, which are, Models, Views, and Controllers. Each of them represents a different part of the system which has its functionalities respectively. Models are the backbone of the system which consists of object classes such as Airport, Physical Runway and Logical Runway. Views consist of mainly the user interfaces and some pop-ups. Finally, controllers act as an interface between Models and Views, helping to update Views while getting data from Models.



*Figure 30 Class diagram*

# 4 Product Testing

## 4.1 Scenario Testing

### 4.1.1 First scenario

User logs in to the system, perform calculation, navigating between visualisations and view calculation breakdown.

#### Acceptance criteria:

User is able to key in inputs and make selections. Calculation is performed with views and table updating, showing the revised parameter values. Calculation breakdown shows how is the revised values obtained from the original parameters.

#### Success scenario:

User key in valid and correct input and obtain the correct results.

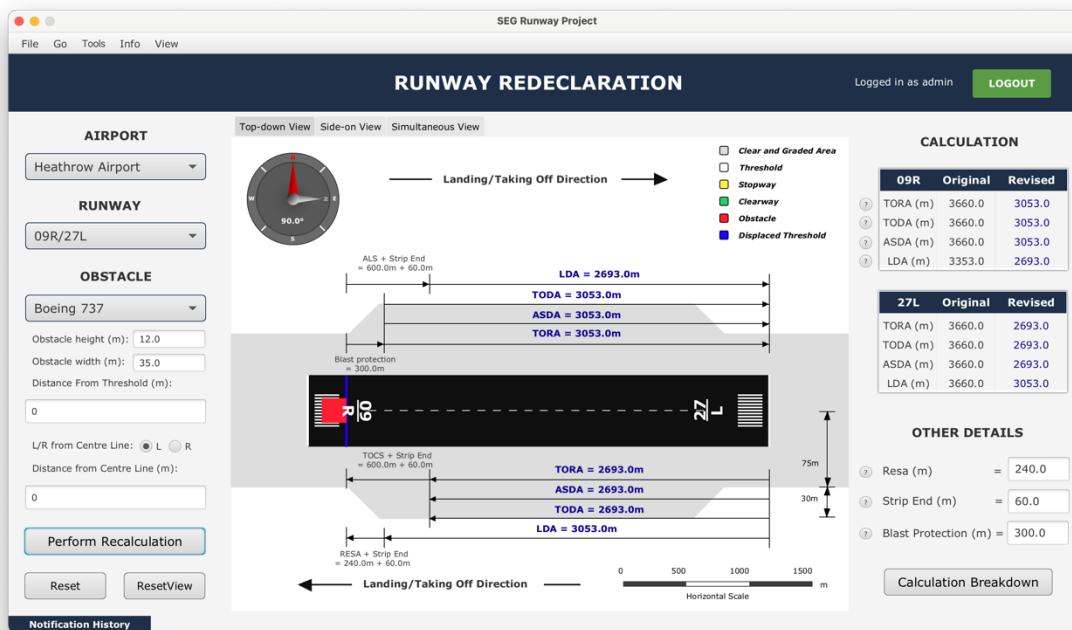


Figure 31 Screenshot showing success scenario where users successfully perform a calculation

Suitable error handlings have been added where error messages are shown for invalid input for distance from threshold, distance from centre line and for other details, including RESA, strip end and blast protection. If no redeclaration is needed, user is also being notified by a pop up and visualisation will show original runway parameters.

## Failed scenario:

### Problem description

For manager and normal air traffic controller, there is an issue of selecting physical runway for the airport. Our idea is to have menu of airports for super admin as he will be able to access all the airports in the system. For the manager and normal air traffic controller, once they are logged in to the system, the airport will be selected automatically based on which airport they are associated to. This causes an issue in loading the physical runway items.

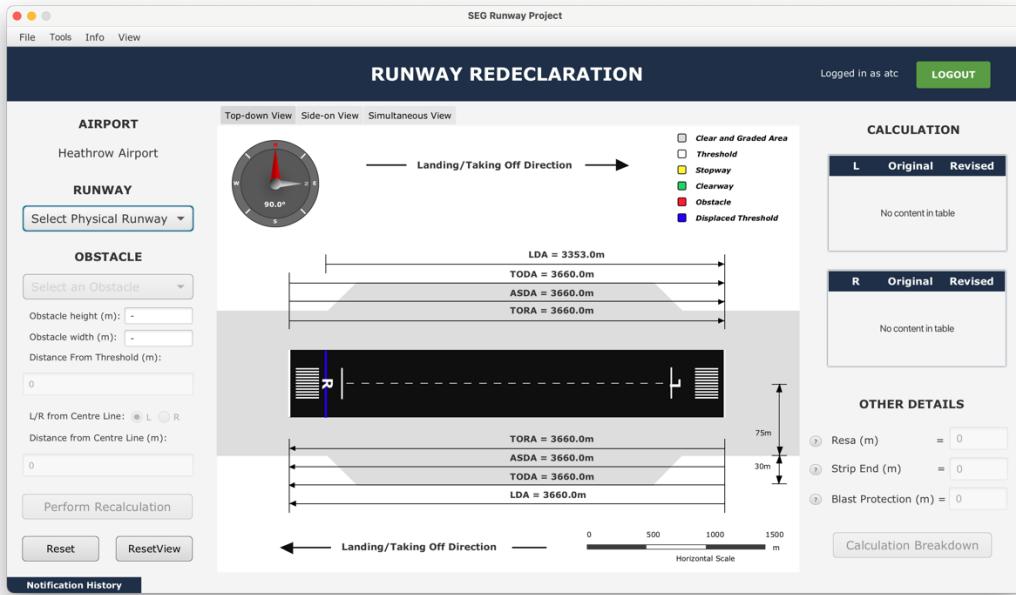


Figure 32 Screenshot showing physical runway item not loaded when clicked on

### Problem identification

The problem is with the function used to load airports, since we have hidden the airport menu for manager and normal users, the event handler for loading physical runway item is not being triggered as the event handlers are associated with the airport menu.

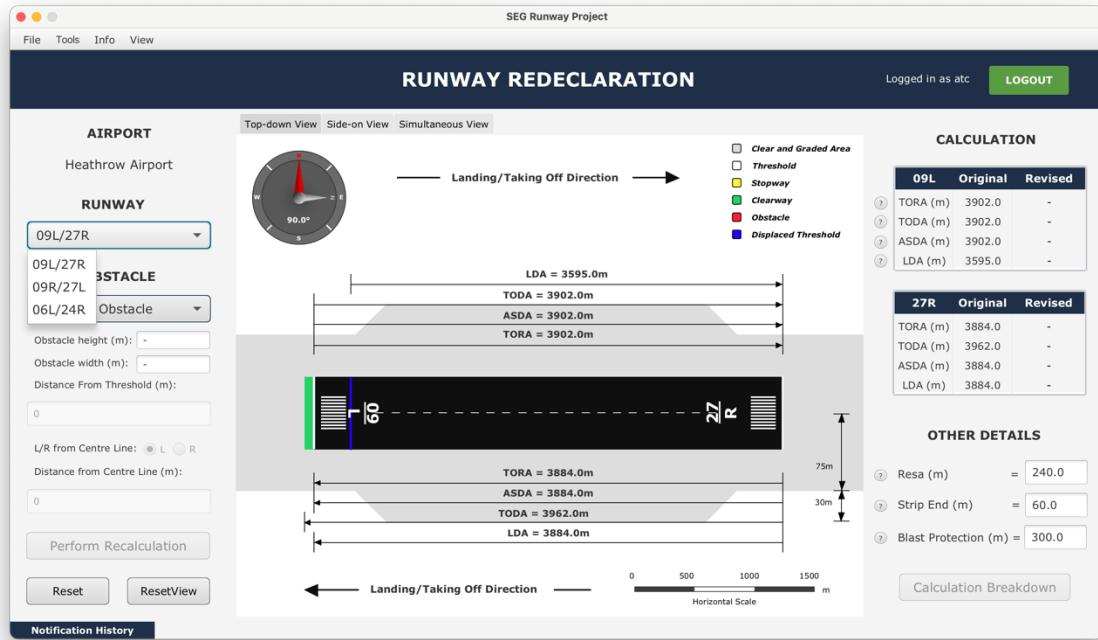
### Solution

The code is modified so that physical runway event handler are extracted as a separate method, if the user is either manager or normal user, we can simply load the physical runways for the particular airports and ignored the airport menu items that have been set invisible.

```

public void addAirportEvent() {
    if(Main.getRole() == 1){
        airportMenuItem.getItems().clear();
        for(Airport airport: airports){
            MenuItem airportMenuItem = new MenuItem(airport.getName());
            airportMenuItem.setStyle("-fx-font-family: Verdana; -fx-font-size: 16px");
            airportMenuItem.setOnAction(e -> {
                resetInactivityTimer();
                airportItem.set(airport);
                physicalRunwayMenu.getItems().clear();
                getAirportMenu().setText(airport.getName());
                physicalRunwayMenu.setText("Select Physical Runway");
                physicalRunwayMenu.setDisable(false);
                performCalculationButton.setDisable(true);
                obstaclesEditing(notAllowed: true);
                addPhysicalRunwayEvent(airport);
            });
        }
        getAirportMenu().getItems().add(airportMenuItem);
    } else{
        addPhysicalRunwayEvent(getAirportSelected());
    }
}

```



The runways can now be loaded for both manager and normal users and they can perform the necessary calculation.

#### 4.1.2 Second scenario

Super admin logs in to the system, tries to add new airport. He wants to edit some airport details and then export the airport information. He also wants to be able to delete airport whenever necessary.

##### Acceptance criteria:

User is able to add new airport by uploading XML files in correct format. He is able to edit airport details accordingly, choose to export the airport information in alternative formats or delete an airport.

##### Success scenario:

User is able to import new airport with details in XML files. Error handling has been implemented where users will be notified of if the file content is in the wrong format.

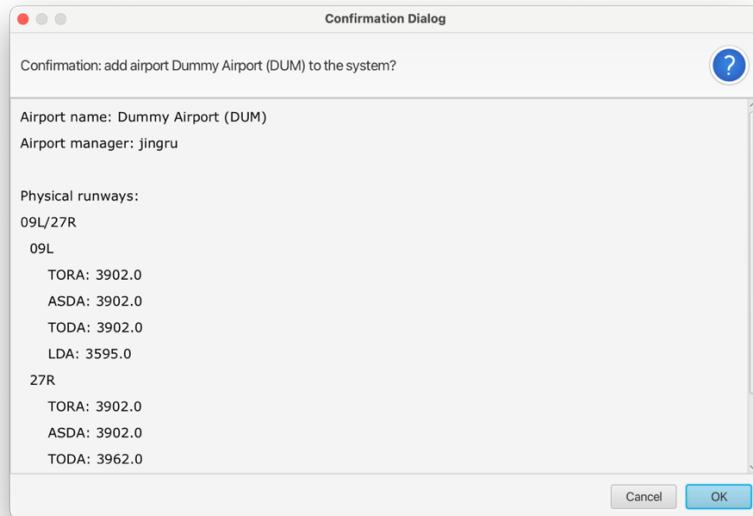


Figure 33 Screenshot showing confirmation dialog when user wants to add new airport

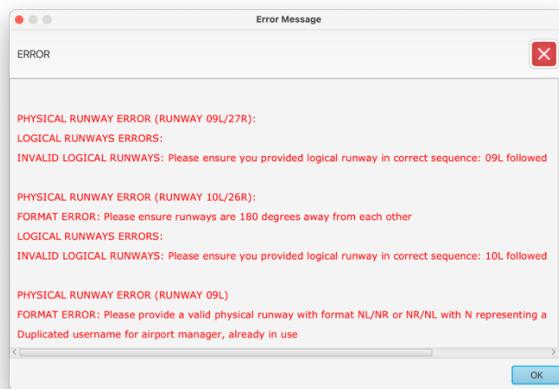
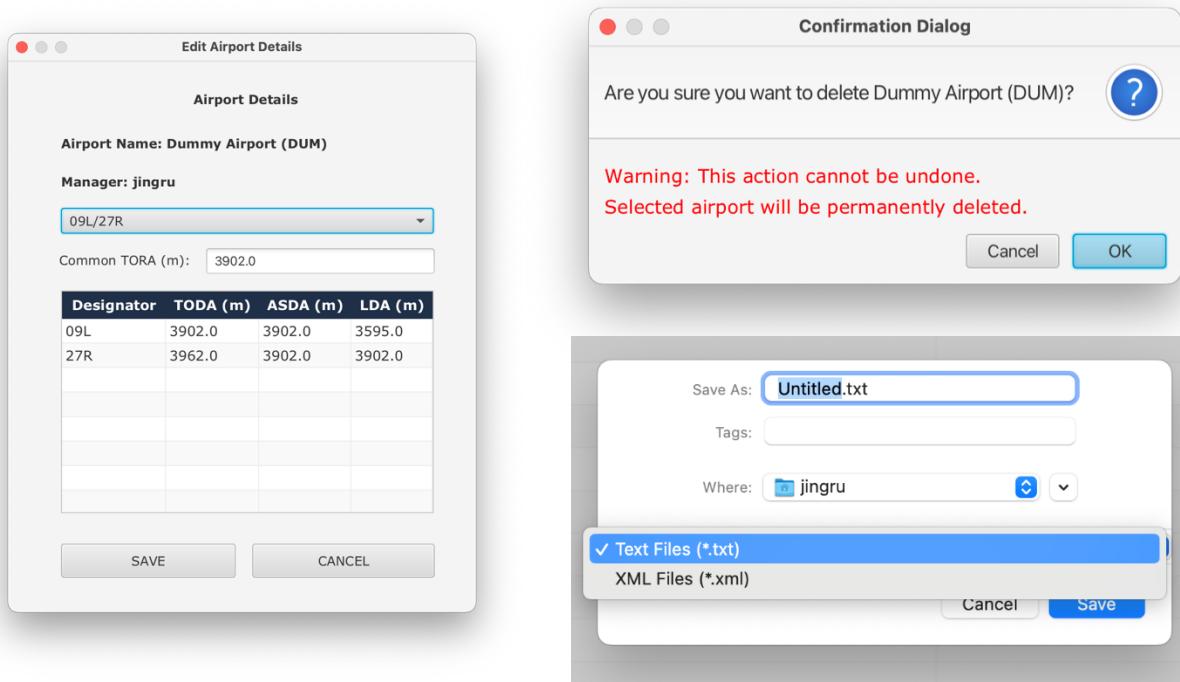


Figure 34 Screenshot showing error message when file is in incorrect XML format

User is able to edit or export airport details, and to delete airport from the system.



#### 4.1.3 Third scenario

Super admin logs in to the system and set up account for manager of the newly added airport. Manager of the airport logs in to the system and add new user account for air traffic controller.

##### Acceptance criteria:

User is able to set up new accounts for normal user or an airport manager.

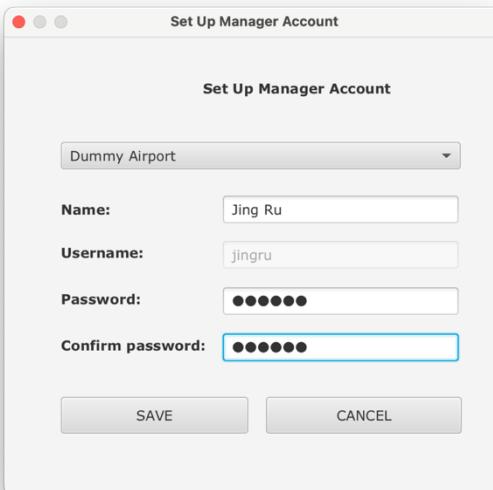


Figure 35 Setting up manager account

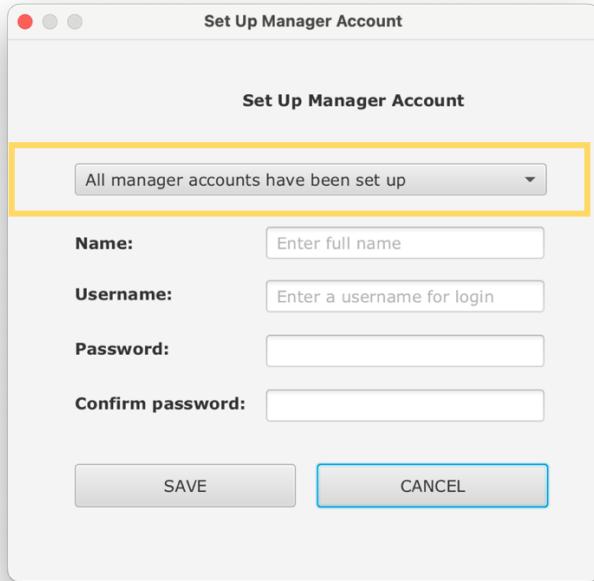


Figure 36 Menu item notifying user that all manager accounts have been set up

The same procedure is carried out successfully by manager to add new users to the airport. In addition, manager can also delete a selected user account. We have removed the option for super admin to delete a manager. However, they can edit manager details such as username or name of the manager. Managers and users account will be automatically deleted once the airport is being deleted.

The screenshot shows a web application titled "SEG Runway Project" with a dark header bar. The header includes "File", "Go", "Info", "Logged in as admin", and a "LOGOUT" button. Below the header is a search bar labeled "Search User". The main content area is titled "USER MANAGER". It features a table with columns: "USERNAME", "NAME", and "AIRPORT". The table contains two rows: one for "manager" (Name: "Manager", Airport: "Heathrow Airport") and one for "jingruu" (Name: "Jing Ruu", Airport: "Dummy Airport"). A yellow box highlights the "NAME" field for the "jingruu" row. At the bottom of the page is a "Set Up Manager Account" button.

USERNAME	NAME	AIRPORT
manager	Manager	Heathrow Airport
jingruu	Jing Ruu	Dummy Airport

Figure 37 Screenshot showing super admin editing username and name for airport manager

## 4.2 Integration Testing

We have been doing integration testing for the main interface since the last code increment. In this increment, we have added two new interfaces, namely the airport manager and the user manager. The main priority of integration testing in this increment is to ensure all information is correctly updated when switching between interfaces.

### 4.2.1 Success Scenarios

1. System updating layout/accessibility accordingly based on user role.

For example,

- Airport as menu button if logged in as super admin, as text displaying airport name if logged in as manager/air traffic controller
  - Menu bar items updating based on user role, e.g. if logged in as super admin, The menu bar will have option to navigate to airport manager and user manager, but as an air traffic controller, there will not be any option to navigate to those interfaces as they do not have access to them based on the access control rules.
2. List of airports updated based on addition/deletion in airport manager. For example, when user is performing calculation in main interface, they choose to navigate to airport manager midway and add new airport/delete existing airport. When they navigate back to the main interface, the previous status is retained but with airport menu updated with new list of airports.

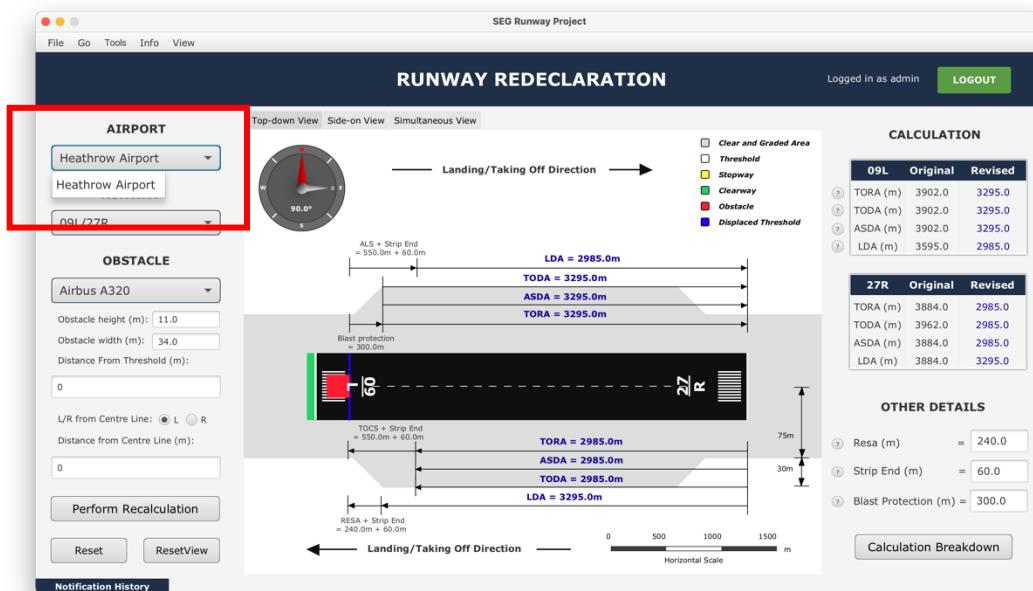


Figure 38 Main interface with original list of airports and user is performing calculation

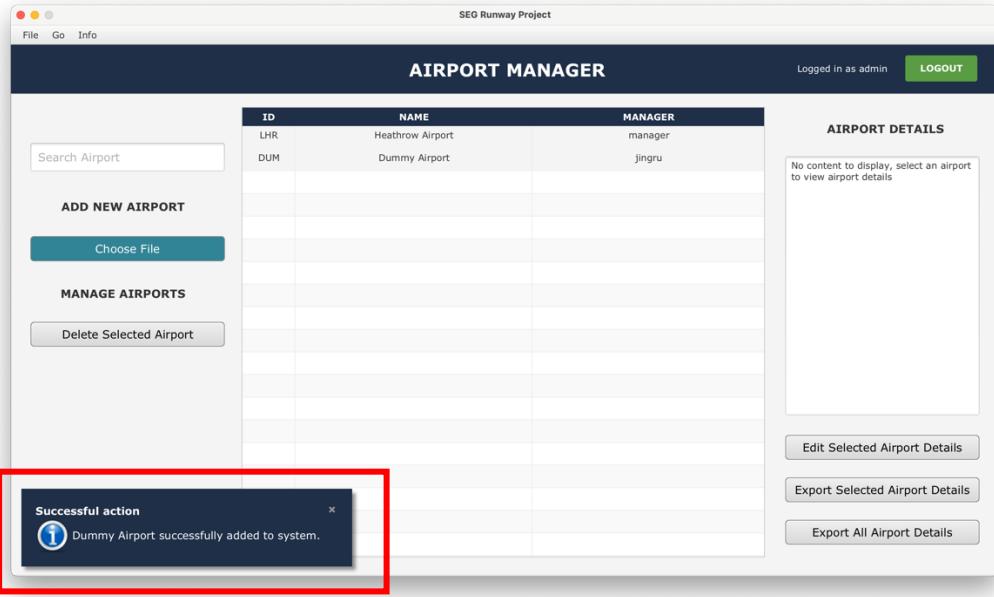


Figure 39 User navigating to airport manager and add a new airport

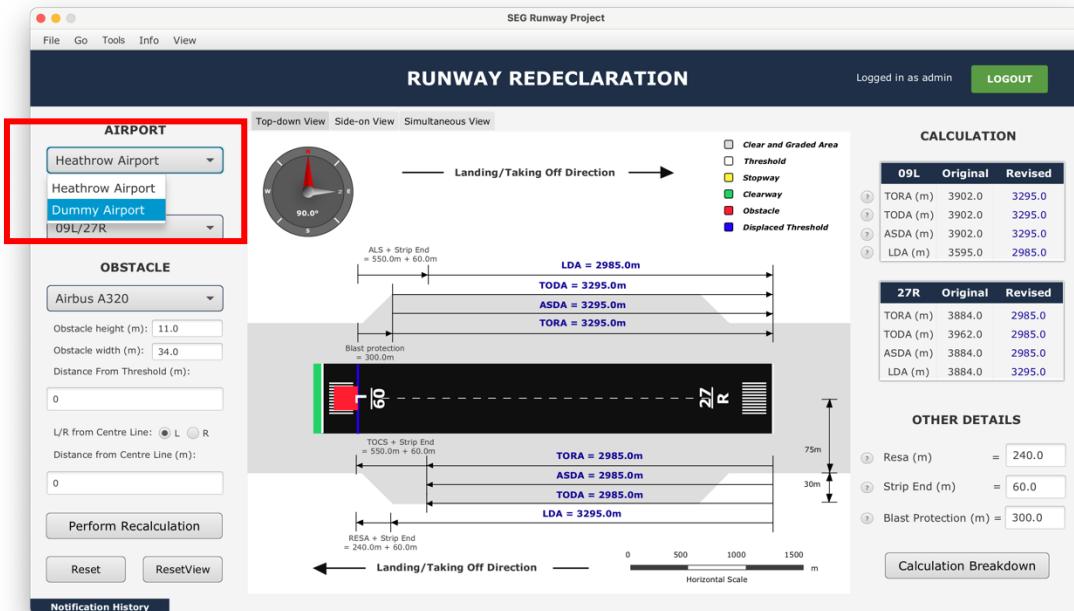


Figure 40 User go back to main interface with airport list updated

3. The added airport are also reflected in user manager where immediately after adding a new airport, when user navigates to the user manager, they are able to see the newly added airport in selection when they try to set up a new manager account.
4. Users added or deleted in user manager is immediately reflected in the system. Once added, user can immediately log in with the new user account.

## 4.2.2 Failed Scenarios

### Problem description

When user had perform calculation or have selected airport or physical runway item, after they log out and log in as another user, the runway is calibrated based on the previous selection before logout even though user has not perform any actions.

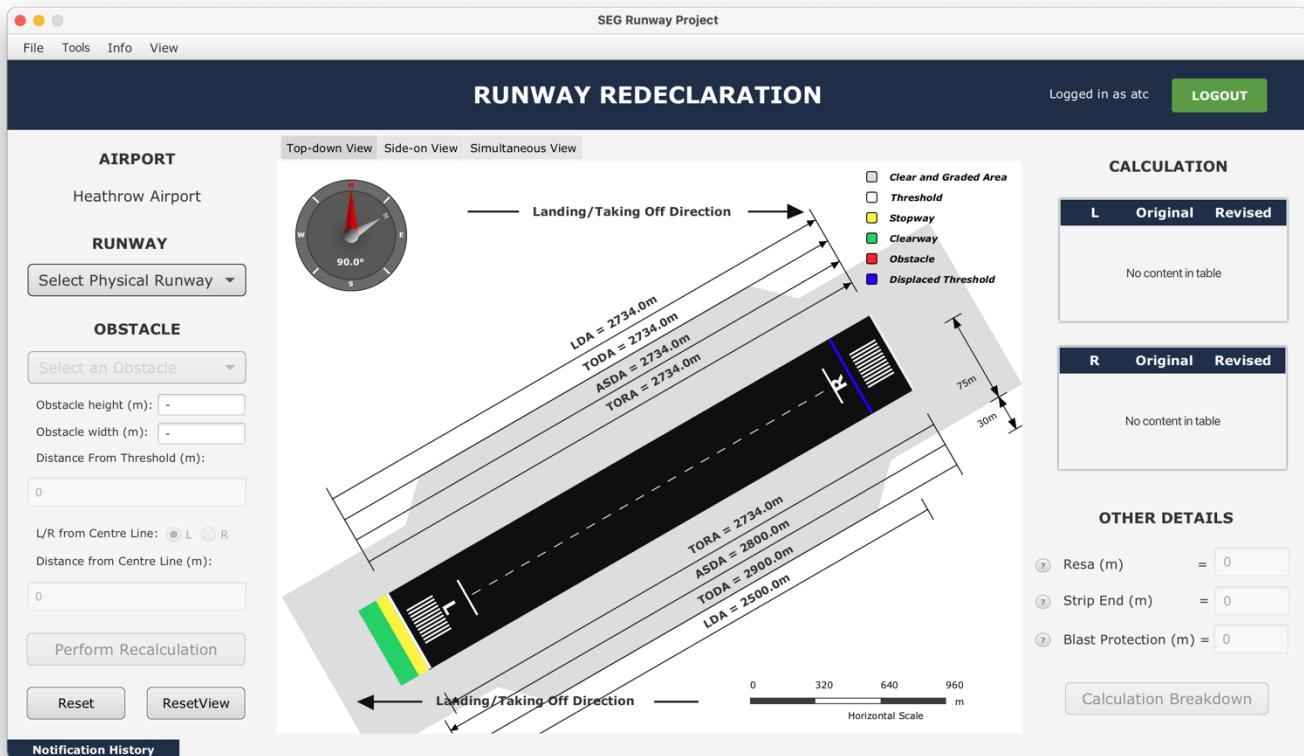


Figure 41 Screenshot showing runway being rotated and showing parameter value even though no runway has been selected

### Problem Identification

The problem is with the part of the code that handles logout, the code simply closes the window and opens the main interface. However, the information in the main controller have not been reset properly and so, data such as airport selected and physical runway selected is kept and used when new user logs in.

### Solution

Added a few lines of code in the function to reset the data similar to the one in reset function. The system is started anew now and problem is solved.

## 5 Third Sprint Review

### 5.1 Sprint Burn-down Chart

**Table 1.1 Tasks for Third Sprint:** List of tasks for the third sprint with effort points determined using planning poker

No.	Tasks	Effort (pts)
	<b>Application - Functionalities</b>	
1.	Allow users to add new airports through GUI form or XML files	16
2.	Allow user to edit existing obstacle details (height and width)	16
3.	Allow users to export airport details in XML files	4
4.	Allow users to export obstacle details in XML files	4
5.	Improve on GUI design using CSS	32
6.	Implement login interface	32
7.	Add more information in help and documentation	16
8.	Automatically rotate runway strip to match compass heading	32
9.	Extend notification features	16
	<b>Planning and Report Writing</b>	
10.	Using UML diagrams, and storyboards to support design decisions	16
11.	Construction of burndown chart for the third increment based on actual progress	4
12.	Response to feedback	4
13.	Application demo and screenshot	8
	<b>Testing and Error Handling</b>	
14.	Checking for invalid inputs (in airport XML)	16
	<b>Submission</b>	
15.	Finalizing submission folder, exporting .jar file, writing the readme	4

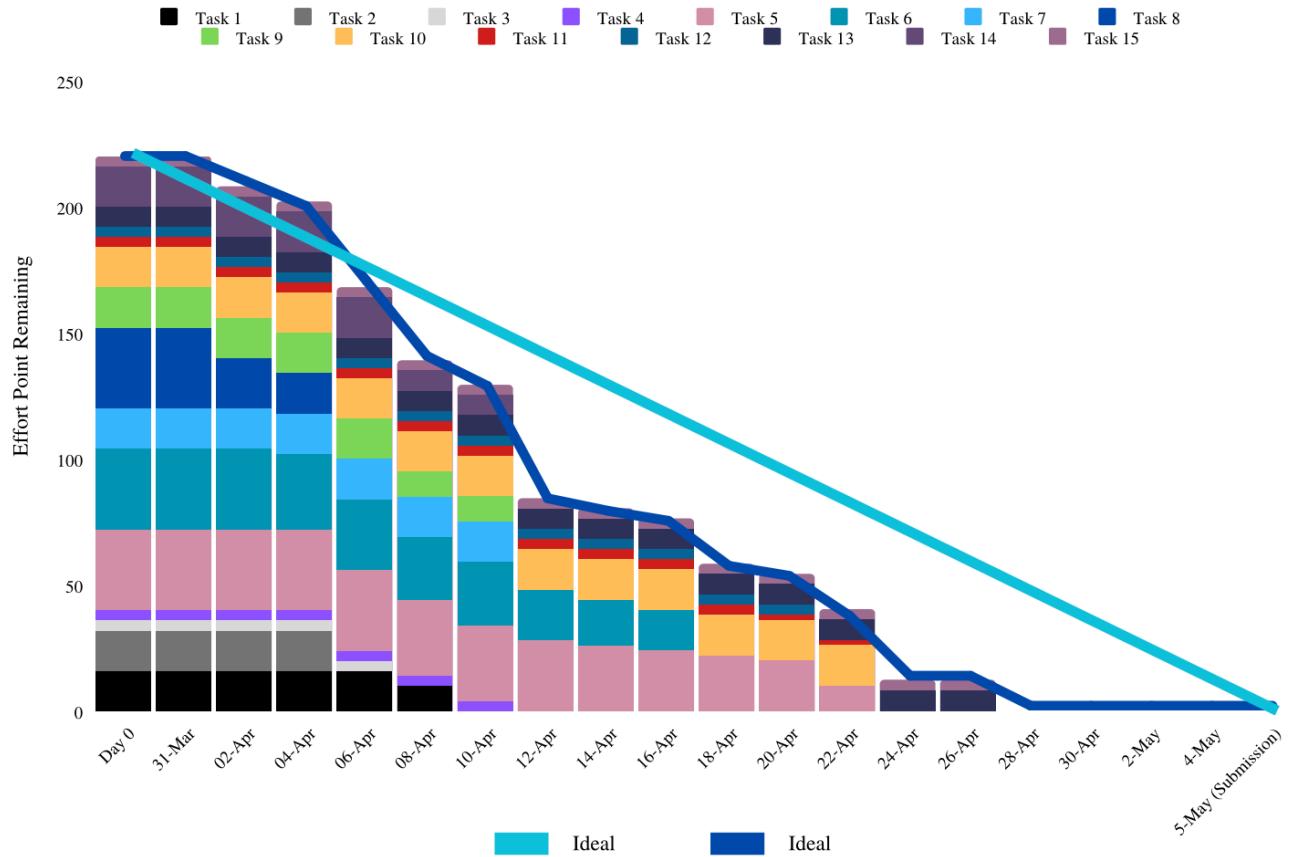


Figure 42 Diagram showing the burndown chart for sprint 3

## 6 Conclusion

Thanks to the generous deadline, the team was able to complete the majority of the tasks that were planned for sprint 3. Given this extra time, the team has made a strategic decision to continue working on extensions for the application. These extensions are aimed at adding further value to the product, thus improving its overall quality and usability.